RLM series

User Guide
RS232 commands

BARCO

Visibly yours

TABLE OF CONTENTS

1. SERIAL COMMUNICATIONS BASICS

Overview

- Communication Protocol
- Data words
- Communication settings
- Hardware
- Syntax
- Serial comms commands

1.1 Communication Protocol

Communication protocol summary

Like every communication method the serial communication uses a particular protocol (ANSI) which must be respected in order to allow communication to take place, following table gives a summary of the predefined communication terms.

Start byte	\xfe
Projector address	
Command byte(s)	
Data bytes[1]	
Checksum byte	
Stop byte	\xff

ANSI

American National standards Institute

Start byte

The "Start byte" informs the projector (in case of transmission) or the computer (in case of reception) that a new data transfer will take place

Projector address

The "projector address" defines the address of the computer the projector wants to talk to (in case of transmission) or the address of the projector that answers (in case of reception).

The maximum number of projectors that can be addressed by one computer is 256.

Command byte(s)

There is at least one command byte to define the action to be performed. Commands that are not often used or complex commands can take more than one byte. All command bytes that are sent by the computer to get information out of the projector are repeated in the answer-data-transfer of the projector.

Data bytes (optional)

Whether the command bytes are followed by one or more data bytes depends on the contents of the command bytes.

Data which contains more than one byte is called a word and can also be signed or unsigned.see "Data words", page 6

---

1. optional

 Some commands do not require a data field(s)

Checksum byte

The "Checksum byte" is used to detect errors during transmission or reception

`Checksum byte = (projectoraddress + command bytes + Data bytes) Modulo \x100`

Stop byte

The "Stop byte" informs the projector (in case of transmission) or the computer (in case of reception) that the data transfer is complete and that the interpretation of the command and data bytes can start.

Acknowledge (ACK)

If the communication link and if the sent commands can be interpreted by the projector an "ACK" command is sent back. In case the commands cannot be interpreted a "NACK" command is sent back.

End of process (EOP)

When the end of process command is activated, an EOP is send after the ACK command to signify the completion of the process.

 for more information about this, see the EOP command.

Note

 Any command byte, data byte or checksum byte that equals `\x80, \xfe, \xff` has to be converted !

1. Transmission
   - instead of \x80, send \x80 followed by \x00
   - instead of \xfe, send \x80 followed by \x7e
   - instead of \xff, send \x80 followed by \x7f
2. Reception
   - replace \x80 followed by \x00 with \x80
   - replace \x80 followed by \x7e with \xfe
   - replace \x80 followed by \x7f with \xff

1.2 Data words

 msb

The **m**ost **s**ignificant **b**yte, is the byte with the greatest weight (value).

 lsb

The **l**ess **s**ignificant **b**yte, is the byte with the smallest weight (value).

Description

A word contains more than one byte, the first byte is the msb, the last byte is the lsb.

Examples of data words

- The projector runtime parameter is described using an unsigned double word where 2 bytes (msb ...lsb) represent the value of the runtime in seconds.

How to calculate the value of a word ?

Lets take the example of a double word returned by the "**projector runtime, read**" commando.

returned data = Data[0]....Data[3]

value = Data[0]*256^3 + Data[1]*256^2 + Data[2]*256 + Data[3].

1.3 Communication settings

Baud rate

Defines the speed of the data transfer. The baud rate can be set using the menu structure.

Consult the Owner's manual of the projector on how to change the baud rate setting.

Data Bits

Eight data bits are used for each character of the data transfer.

Parity

There is no parity bit used to perform error checking.

Stop Bits

One stop bit is used to define the end of a character.

1.4 Hardware

Overview

• Female D9–pin connector labelled "RS232 IN" used to connect the projector with the computer.

Pin out of the D9 connector:

Pin	Name	Full Name
1	CD	Carrier Detect
2	RxD	Received Data
3	TxD	Transmitted Data
4	DTR	Data Terminal Ready
5	GND	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

• Standard serial Cable
  - order number R9827560 (cable length = 15m)
  - order number R9827570 (cable length = 30m)

D9 female			D9 male
1 CD			1 CD
2 RxD			2 RxD
3 TxD			3 TxD
4 DTR			4 DTR
5 GND			5 GND
6 DSR			6 DSR
7 RTS			7 RTS
8 CTS			8 CTS
9 RI			9 RI
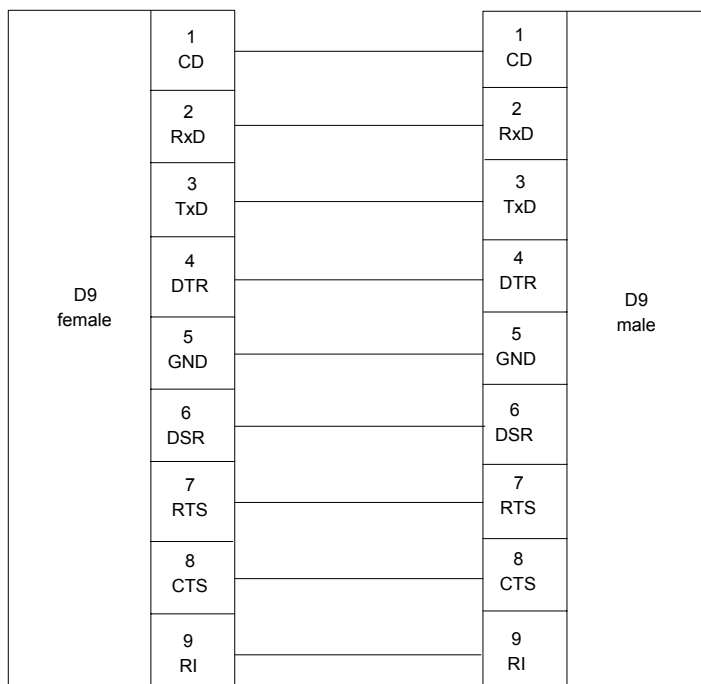
Image 1-1
connection: IBM PC or compatible → projector

DIN mini 8			D9 male
1 HSKo			4 DTR
2 HSKi			8 CTS
3 TxD-			3 TxD
4 GND			5 GND
5 RXD-			2 RxD
6 TXD+			
7 n.c.			
8 RXD+			
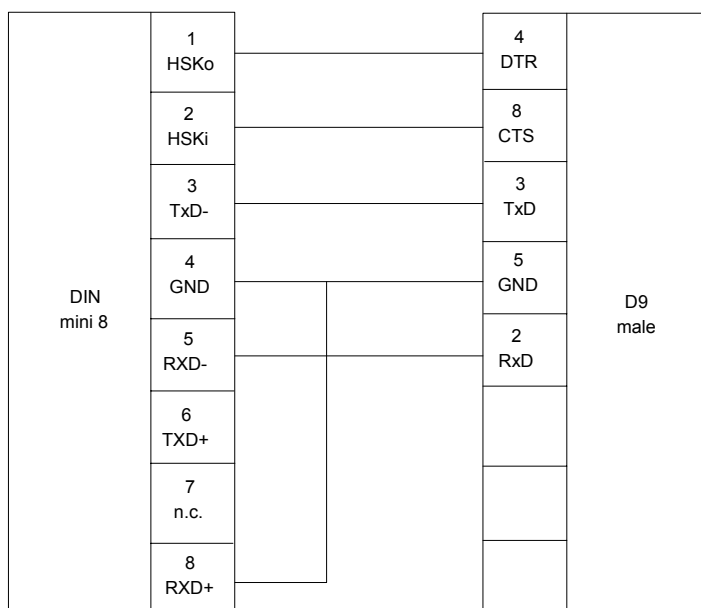
Image 1-2
connection: MAC → projector

Signal levels

Following table gives the signal levels corresponding to the different states:

State	Voltage
Off = 1	-9V
On = 0	+9V

1.5   Syntax

Characters

In this manual, all characters are expressed using the C- language syntax:

decimal values	**ddd**[2]
hexadecimal values	\x**hh**[3]

The characters that can be used are

For text fields: "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+-/&@#_.:;abcdefghijklmnopqrstuvwxyz"
For filenames: "?*ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_ -"

Negative values /numbers

The 2s complement number system is used to express negative numbers

Pascal- language string

A Pascal-language string consists of one or more characters. The first character of the string contains the length of the string. Therefore, a Pascal-language string is limited to 255 characters.

Pascal string 'Hello world"

length	\x0b
'h'	\x68
'e'	\x65
'l'	\x6c
'l'	\x6c
'o'	\x6f
"	\x20
'w'	\x77
'o'	\x6f
'r'	\x72
'l'	\x6c
'd'	\x64

C-language string

A C-language string consists of one or more characters. The last character of the string is always the NULL character \x00. Therefore, the length of a C-language string is determined by the position of the NULL character..

C string 'Hello world"

'h'	\x68
'e'	\x65
'l'	\x6c
'l'	\x6c
'o'	\x6f
"	\x20
'w'	\x77

---

2. ddd=0...255
3. hh=00...ff

'o'	\x6f
'r'	\x72
'l'	\x6c
'd'	\x64
NULL	\x00

> **standard file**
>
> Predefined file stored in read-only memory

> **custom file**
>
> File created by the user and stored in non-volatile read write memory

Filename

A filename is specified as a C-language string. This string has to follow some rules:

x	x	x	x	x	x	x	x	.	y	z	z	NULL

1. length string = 12
2. x = character of the base name formed by max 8 characters
3. y = kind of file (= 1 character)

's'	standard file
'c'	custom file

4. z = file index (= 2 characters: 0..9)
    zz specifies the location in memory where the file is stored.
    -   for standard files: zz = 00...maximum standard files
    -   for custom files: zz = 00...63 where 00 is reserved for the file 'none .c00' (file which is loaded when no signal is applied)
5. yzz is a unique combination. In other words, no two files can exist with the same extension yzz.

> **To specify more than one file you can use the question mark (?) and (*) wildcard character for x, y and z (or any other character).**
>
> **?: This wildcard character can represent any possible character on a particular location.**
>
> *** : This wildcard character can represent any possible character(s) following the wildcard's position in the string.**

-   "ntsc .c01"
-   "svga_60v.s?7"
-   "????????.???"
-   "sv*.*"

1.6   Serial comms commands

Basic Commands

Basic commands are the commands used to control the basic parameters of the projector.

The **Contrast increment** command is a basic command.

Advanced Commands

Advanced commands are used to control advanced parameters of the projector.

The **Pip loop focus** is an advanced parameter

This manual is divided in two parts, the first part contains the basic (elementary) commands while the second part gives an overview of the available advanced commands.

Some commands are illustrated with an example.

2. BASIC COMMANDS

Overview

- End of process (EOP)
- Identification Commands
- RCU commands
- Projector ON/OFF, Write
- Projector's Status, Read
- Lens commands
- Picture setting
- Menu Exit
- Source selection
- Scanned inputs, read
- Scan inputs
- File, Load
- Orientation, Write
- Orientation, Read
- Shutter, Write
- Shutter, Read
- Freeze, Write
- Freeze, Read
- Logo ON/OFF
- Background, Write
- Background, Read
- Aspect ratio, Write
- Aspect Ratio, Read
- Show Native resolution (ON/OFF), Write
- Show Native resolution (ON/OFF), Read
- Full screen representation(ON/OFF), Write
- Full screen representation (ON/OFF), Read
- Keystone, Write
- Keystone, Read
- Lamp commands
- Image files
- Color temperature
- Standby Timer, Write
- Standby Timer, Read
- Switching mode, Write
- Switching mode, Read
- Filmmode detection, Increment/Decrement
- Filmmode detection, Write
- Filmmode detection, Read
- Autoimage execute
- Blanking

2.1 End of process (EOP)

Description

When activated returns an EOP whenever a command has been executed (processed).

An activation byte has to be sent in order to activate or disactivate the end of process function.

Command

Command [0]	\x00
Command [1]	\x01

Sent Data

Data[0] = activation byte : \x00 = OFF ; \x01 = ON

Returned data

When you send a particular command (for example projector ON), the EOP will be returned when the command has been executed

Command [0]	\x00
Command [1]	\x01

> **The EOP will be sent after the ACK.**
>
> **In case of a read command there is no EOP returned by the projector.**

2.2 Identification Commands

2.2.1 Device type, Read

Description

Reads the device (projector) type, the returned string is a Pascal-language string.

Command

Command [0]	\x6b

Returned Data

Data[0] = byLength = 1 byte describing the length of the string.

Data[1] = strDeviceType = pascal-language string describing the device type.

> **An example of device type is "RLM"**

2.2.2 Projector's Software type, Read

Description

Reads the Projector's Software type, the returned string is a Pascal-language string.

Command

Command [0]	\x6a

Returned Data

Data[0] = byLength = 1 byte describing the length of the software version string.

Data[1] = strSoftwareType = pascal-language string describing the software type.

> **An example of software type is "Standard"**

2.2.3 Projector's Software version, Read

Description

Reads the Projector's Software version, the returned string is a Pascal-language string.

Command

Command [0]	\x60

Returned Data

Data[0] = byLength = 1 byte describing the length of the software version string.

Data[1] = strSoftwareVersion

2.2.4 Projector's Serial number, Read

Description

Reads the Projector's serial number, the returned string is a Pascal-language string.

Command

Command [0]	\x61

Returned Data

Data[0] = \x07 = length of the serial number = 7 characters

Data[1] = string of length 7

2.2.5 Projector's runtime, Read

Description

Reads the Projector's runtime in seconds

Command

Command [0]	\x62

Returned Data

unsigned double word (4 bytes).

Data[0] = msb

Data[1]

Data[2]

Data[3] = lsb

2.2.6 Projector address, Write

Description

Writes a new address to the projector

Command

Command [0]	\x6D

Data

Data[0] = unsigned byte

Example

Writes a new address to the projector, being "1"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x6d
Data[0]	\x01
Checksum	\x6f
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.2.7   Projector language, Read

Description

Reads the active language for the projector

Command

Command [0]	\x71

Returned Data

Data[0] = unsigned byte

byte value	language
0	English
1	Français
2	Deutsch
3	Español
4	Japanese
5	Chinese

Example

reads the active language, being "Français"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x71
Checksum	\x72
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x71
Data[0]	\x01
Checksum	\x73
Stop	\xff

2.2.8   Projector language, Write

Description

Sets a new language for the projector

Command

Command [0]	\x70

Data

Data[0] = unsigned byte

byte value	language
0	English
1	Français
2	Deutsch
3	Español
4	Japanese
5	Chinese

Example

Sets the language to "Français"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x70
Data[0]	\x01

Transmit	
Checksum	\x72
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.2.9   Baudrate, Write

Description

Sets the baudrate

Command

Command [0]	\x75

Data

string = C-language string

possible string values
9600
19200
38400
57600
115200

2.3   RCU commands

Description

Simulation of the RCU codes sent by the RCU remote control.

For all commands, see "RCU commands", page 165.

Command

Command [0]	\x30

Data

Data[0] = byIRcode = 1 byte representing the RC5 code of the function to be executed.

Wrepeat (optional) = word (2 bytes) representing the duration.

2.4   **Projector ON/OFF, Write**

Description

Switching ON/OFF the projector

Command

Switching the projector ON

Command [0]	\x65

Switching the projector OFF

Command [0]	\x66

Example

Switches the projector ON

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x65
Checksum	\x66
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.5   **Projector's Status, Read**

Description

Reads the Projector's Status

Command

Command [0]	\x67

Returned Data

Data [0] = unsigned byte

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Not used	Not used	Cool down cycle active = 0 inactive = 1	Not used	Not used	Not used	Textbox ON = 1 OFF = 0	Projector ON = 1 OFF = 0

> **bit5 gives an indication on whether the projector can be (re)started or not (hot restrike), if the cool down cycle is active (bit5 = 0) the projector can not be restarted until the cool down cycle has been completed.**

Example

Read the status being : `\x23`:

- projector is ON
- textbox is on
- cool down cycle is inactive

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x67`
Checksum	`\x68`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Awnser)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x67`
Data[0]	`\x23`
Checksum	`\x8b`
Stop	`\xff`

2.6 Lens commands

2.6.1 Lens shift

Description

Shifts the lens down or up depending on the data sent.

Command

Command [0]	`\xf4`
Command [1]	`\x81`

Data

Data[0] = byDirection = 1 byte, gives the shift direction:

0 = down ; 1 = up; 2 = left; 3 = right

2.6.2 Lens Zoom

Description

To Zoom the image in or out depending on the data sent.

Command

Command [0]	\xf4
Command [1]	\x82

Data

Data[0] = byDirection = 1 byte, gives the Zoom direction (in/out):

0 = out ; 1 = in

2.6.3 Lens Focus

Description

To Focus the image.

Command

Command [0]	\xf4
Command [1]	\x83

Data

Data[0] = byDirection = 1 byte : 0 = "-" ; 1 = "+"

2.7 Picture setting

2.7.1 Contrast, Increment

Description

Increments the contrast by one.

Command

Command [0]	\x22
Command [1]	\x01

Example

Increment contrast of a projector with address \x01 by one.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x22
Command[1]	\x01

Transmit	
Checksum	\x24
Stop	\xff

2.7.2 Contrast, decrement

Description

Decrements the contrast by one.

Command

Command [0]	\x23
Command [1]	\x01

2.7.3 Contrast, write

Description

Writes a new contrast value to the projector

Command

Command [0]	\x20
Command [1]	\x01

Data

Data [0] = contrast value

	Data[0] = unsigned byte

Example

Writes a new contrast value of \x20 to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x01
Data	\x20
Checksum	\x42
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

2.7.4   Contrast, Read

Description

Reads the actual contrast value of the projector

Command

Command [0]	\x21
Command [1]	\x01

Returned Data

Data [0] = contrast value

Example

Reads the actual contrast value of the projector, the awnser being \x20

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x01
Checksum	\x23
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x01
Data [0]	\x20
Checksum	\x43
Stop	\xff

2.7.5 Minimum Contrast, Read

Description

Reads the minimum (range) contrast value of the projector

Command

Command [0]	\x26
Command [1]	\x01

Returned Data

Data [0] = minimum contrast value

Example

Reads the minimum contrast value of the projector, the awnser being \x00

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x01
Checksum	\x28
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x01
Data [0]	\x00
Checksum	\x28
Stop	\xff

2.7.6 Maximum Contrast, Read

Description

Reads the Maximum (range) contrast value of the projector

Command

Command [0]	\x27
Command [1]	\x01

Returned Data

Data [0] = Maximum contrast value

Example

Reads the Maximum contrast value of the projector, the awnser being \x64

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x01
Checksum	\x29
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x01
Data [0]	\x64
Checksum	\x8d
Stop	\xff

2.7.7 Brightness, increment

Description

Increments the Brightness by one.

Command

Command [0]	\x22
Command [1]	\x02

2.7.8 Brightness decrement

Description

Decrements the Brightness by one.

Command

Command [0]	\x23
Command [1]	\x02

2.7.9 Brightness, Write

Description

Writes a new Brightness value to the projector

Command

Command [0]	\x20
Command [1]	\x02

Data

Data [0] = contrast value

value = unsigned byte

Example

Writes a new Brightness value of \x20 to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x02
Data	\x20
Checksum	\x43
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.7.10 Brightness, Read

Description

Reads the actual Brightness value of the projector

Command

Command [0]	\x21
Command [1]	\x02

Returned Data

Data [0] = Brightness value

Example

Reads the actual Brightness value of the projector, the awnser being \x20

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x02
Checksum	\x24
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x02
Data [0]	\x20
Checksum	\x44
Stop	\xff

2.7.11 Minimum Brightness, Read

Description

Reads the minimum (range) Brightness value of the projector

Command

Command [0]	\x26
Command [1]	\x02

Returned Data

Data [0] = minimum Brightness value

Example

Reads the minimum Brightness value of the projector, the awnser being \x00

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x02
Checksum	\x29
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x02
Data [0]	\x00
Checksum	\x29
Stop	\xff

2.7.12 Maximum Brightness, Read

Description

Reads the Maximum (range) Brightness value of the projector

Command

Command [0]	\x27
Command [1]	\x02

Returned Data

Data [0] = Maximum Brightness value

Example

Reads the Maximum Brightness value of the projector, the awnser being `\x64`

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x27`
Command[1]	`\x02`
Checksum	`\x30`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Answer)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x27`
Command[1]	`\x02`
Data [0]	`\x64`
Checksum	`\x8e`
Stop	`\xff`

2.7.13   Saturation increment

Description

Increments the saturation by one.

Command

Command [0]	`\x22`
Command [1]	`\x03`

2.7.14   Saturation, decrement

Description

Decrements the Saturation by one.

Command

Command [0]	\x23
Command [1]	\x03

2.7.15 Write Saturation

Description

Writes a new Saturation value to the projector

Command

Command [0]	\x20
Command [1]	\x03

Data

Data [0] = Saturation value

value = unsigned byte

Example

Writes a new Brightness value of \x20 to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x03
Data	\x20
Checksum	\x44
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.7.16 Saturation, Read

Description

Reads the actual Saturation value of the projector

Command

Command [0]	\x21
Command [1]	\x03

Returned Data

Data [0] =Saturation value

Example

Reads the actual Saturation value of the projector, the awnser being \x20

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x03
Checksum	\x25
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x03
Data [0]	\x20
Checksum	\x45
Stop	\xff

2.7.17 Minimum Saturation, Read

Description

Reads the minimum (range) Saturation value of the projector

Command

Command [0]	\x26
Command [1]	\x03

Returned Data

Data [0] = minimum Saturation value

Example

Reads the minimum Saturation value of the projector, the awnser being \x00

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x03
Checksum	\x30
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x03
Data [0]	\x00
Checksum	\x30
Stop	\xff

2.7.18 Maximum Saturation, Read

Description

Reads the Maximum (range) Saturation value of the projector

Command

Command [0]	\x27
Command [1]	\x03

Returned Data

Data [0] = Maximum Saturation value

Example

Reads the Maximum Saturation value of the projector, the awnser being \x7f

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x03
Checksum	\x31
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x03
Data [0]	\x7f
Checksum	\xaa
Stop	\xff

2.7.19   Tint, increment

Description

Increments the Tint by one.

Command

Command [0]	\x22
Command [1]	\x04

2.7.20   Tint, decrement

Description

Decrements the Tint by one.

Command

Command [0]	\x23
Command [1]	\x04

2.7.21 Tint, Write

Description

Writes a new Tint value to the projector

Command

Command [0]	\x20
Command [1]	\x04

Data

Data [0] = msb

Data [1] = lsb

type = signed word

Example

Writes a new Tint value of \x20 to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x04
Data[0]	\x00
Data[1]	\x20
Checksum	\x45
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.7.22 Tint, Read

Description

Reads the actual Tint value of the projector

Command

Command [0]	\x21
Command [1]	\x04

Returned Data

signed word :

Data [0] = msb

Data [1] = lsb

Example

Reads the actual Tint value of the projector, the awnser being： `\x00 \x20`

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x04`
Checksum	`\x26`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Answer)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x04`
Data [0]	`\x00`
Data [1]	`\x20`
Checksum	`\x46`
Stop	`\xff`

2.7.23   Minimum Tint, Read

Description

Reads the minimum (range) Tint value of the projector

Command

Command [0]	`\x26`
Command [1]	`\x04`

Returned Data

signed word :

Data [0] = msb

Data [1] = lsb

Example

Reads the minimum Tint value of the projector, the awnser being \xff\x4c

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x04
Checksum	\x31
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x04
Data [0]	\xff
Data [1]	\x4c
Checksum	\x76
Stop	\xff

2.7.24   Maximum Tint, Read

Description

Reads the Maximum (range) Tint value of the projector

Command

Command [0]	\x27
Command [1]	\x04

Returned Data

signed word :

Data [0] = msb

Data [1] = lsb

Example

Reads the Maximum Tint value of the projector, the awnser being `\x00\xb4`

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x27`
Command[1]	`\x04`
Checksum	`\x32`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Answer)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x27`
Command[1]	`\x04`
Data [0]	`\x00`
Data [1]	`\xb4`
Checksum	`\xe0`
Stop	`\xff`

2.7.25 Gamma, increment

Description

Increments the Gamma by one.

Command

Command [0]	`\x22`
Command [1]	`\x70`

2.7.26 Gamma, decrement

Description

Decrements the Gamma by one.

Command

Command [0]	\x23
Command [1]	\x70

2.7.27 Gamma, Write

Description

Writes a new Gamma value to the projector

Command

Command [0]	\x20
Command [1]	\x70

Data

Data [0] = Gamma value

	Data[0] = unsigned byte

Example

Writes a new Gamma value of \x11 to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x70
Data	\x11
Checksum	\xa2
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.7.28 Gamma, Read

Description

Reads the actual Gamma value of the projector

Command

Command [0]	\x21
Command [1]	\x70

Returned Data

Data [0] = Gamma value

Example

Reads the actual Gamma value of the projector, the awnser being \x11

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x70
Checksum	\x92
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x70
Data [0]	\x11
Checksum	\xa3
Stop	\xff

2.7.29 Minimum Gamma, Read

Description

Reads the minimum (range) Gamma value of the projector

Command

Command [0]	\x26
Command [1]	\x70

Returned Data

Data [0] = minimum Gamma value

Example

Reads the minimum Gamma value of the projector, the awnser being `\x10`

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x26`
Command[1]	`\x70`
Checksum	`\x97`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Answer)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x26`
Command[1]	`\x70`
Data [0]	`\x10`
Checksum	`\xa7`
Stop	`\xff`

2.7.30 Maximum Gamma, Read

Description

Reads the Maximum (range) Gamma value of the projector

Command

Command [0]	`\x27`
Command [1]	`\x70`

Returned Data

Data [0] = Maximum Gamma value

Example

Reads the Maximum Gamma value of the projector, the awnser being `\x1c`

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x70
Checksum	\x98
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x70
Data [0]	\x1c
Checksum	\xb4
Stop	\xff

2.7.31 Phase, increment

Description

Increments the Phase by one.

Command

Command [0]	\x22
Command [1]	\x06

2.7.32 Phase, decrement

Description

Decrements the Phase by one.

Command

Command [0]	\x23
Command [1]	\x06

2.7.33   Phase, Write

Description

Writes a new Phase value to the projector

Command

Command [0]	\x20
Command [1]	\x06

Data

Data [0] = Phase value

---

value = unsigned byte

Example

Writes a new Phase value of `\x1a` to the projector

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x06
Data	\x1a
Checksum	\x41
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.7.34   Phase, Read

Description

Reads the actual Phase value of the projector

Command

Command [0]	\x21
Command [1]	\x06

Returned Data

Data [0] = Phase value

Example

Reads the actual Phase value of the projector, the awnser being `\x1a`

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x06`
Checksum	`\x28`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Answer)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x06`
Data [0]	`\x1a`
Checksum	`\x42`
Stop	`\xff`

2.7.35 Minimum Phase, Read

Description

Reads the minimum (range) Phase value of the projector

Command

Command [0]	`\x26`
Command [1]	`\x06`

Returned Data

Data [0] = minimum Phase value

Example

Reads the minimum Phase value of the projector, the awnser being `\x00`

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x06
Checksum	\x33
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x26
Command[1]	\x06
Data [0]	\x00
Checksum	\x33
Stop	\xff

2.7.36 Maximum Phase, Read

Description

Reads the Maximum (range) Phase value of the projector

Command

Command [0]	\x27
Command [1]	\x06

Returned Data

Data [0] = Maximum Phase value

Example

Reads the Maximum Phase value of the projector, the awnser being \x1f

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x27

Transmit	
Command[1]	\x06
Checksum	\x34
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x27
Command[1]	\x06
Data [0]	\x1f
Checksum	\x4d
Stop	\xff

2.8   Menu Exit

Description

Exits all the menus (goes back to the operational mode)

Command

Command [0]	\x42
Command [1]	\x01

Example

Exits **all** the menu of a projector with address \x01.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x42
Command[1]	\x01
Checksum	\x44
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.9   Source selection

Description

Selects an input source.

Command

Command [0]	\x38

Data

Data[0] = bySource, byte describing the source to be linked to the window.

Byte value	source
1	BNC's
2	D15
3	Composite video
4	SVideo
5	DVI
6	Firewire
7	SDI
8	IQ PC

For more info about the input configurations, see the projector's Owner's manual.

Data[1] = byMode

for BNC's and D15: the data byte gives the type of signal present on the BNC's/D15

Byte value	Mode
0	RGBHV
1	RGBCS (analog[4])
2	RGBCS (digital)
3	RG_sB
100 (x\64)	PrYPb
101 (x\65)	RGB video
102 (x\66)	RG_SB video

---

4. the signal present on the Cs signal is a composite video signal containing the composite sync

for composite video: the data byte describes the location of the composite video source signal to be selected.

Byte value	Mode
0	Video BNC
1	Vertical sync BNC
2	Red BNC
3	Green BNC
4	Blue BNC
5	SVideo Y
6	SVideo C
7	Video 2[5]

for Svideo: the data byte describes the location of the S-Video signal to be selected.

Byte value	Mode
100 (x\64)	SVideo 1
101 (x\65)	SVideo on Vs and Video BNC
102 (x\66)	SVideo on R & B
103 (x\67)	SVideo 4[5]

If byMode is ommited the mode is automaticaly detected.

2.10 Scanned inputs, read

Description

read the scanned inputs and returns the type of source signal present .

Command

Command [0]	\x39
Command[1]	\x01

Returned Data

10 unsigned words.see chapter "1. Serial communications Basics", "Data words", page 6

Data word	scanned input	possible sources, Wvalue =
wValue1	BNC (RED)	0 = none ; 1 = RED; 2 = Pr ; 3 = S-VID(chr) ; 4 = VideoR
wValue2	BNC (GREEN)	0 = none ; 1 = GREEN ; 2 = G_s ; 3 = Y ; 4 = VideoG
wValue3	BNC (BLUE)	0 = none ; 1 = BLUE ; 2 = Pb ; 3 = S-VID(lum) ; 4 = VideoB
wValue4	BNC (HS)	0 = none ; 1 = HS; 2 =CS
wValue5	BNC (VS)	0 = none ; 1 = VS; 2 = S-VID(chr) ; 3 = Video VS
wValue6	BNC (VIDEO)	0 = none ; 1 = VIDEO ; 3 = S-VID(lum)

---

5. optional

Data word	scanned input	possible sources, Wvalue =
wValue7	S-VIDEO (lum)	0 = none ; 1 = S-VID(lum) ; 2 = Video Y
wValue8	S-VIDEO (chr)	0 = none ; 1 = S-VID(chr) ; 2 = VideoC
wValue9	DVI	0 = none ; 1 = DVI
wValue10	PC	0 = none ; 1 = RGBHV ; 2 = RG$_s$B

For more info about the input configurations, see the projector's Owner's manual.

2.11 Scan inputs

Description

Scans the inputs, the results (sources) can be read using the "scanned inputs" read command.

Command

Command [0]	\x39
Command[1]	\x02
Command[2]	\x00

2.12 File, Load

Description

Loads a specific file

Command

Command [0]	\xbd
Command [1]	\x82

Data

File name, no wild cards allowed.

Example

Load a file named

test.c05

on projector with address 01.

Transmit	
Start	\xfe
Projector Address	\x01
Command [0]	\xbd
Command [1]	\x82
Data [0]	\x74 (='t')
Data [1]	\x65 (='e')
Data [2]	\x73 (='s')
Data [3]	\x74 (='t')

Transmit	
Data [4]	\x2e (='.')
Data [5]	\x63 (='c')
Data [6]	\x30 (='0')
Data [7]	\x35 (='5')
Data [8]	\x00
Checksum	\xF6
Stop	\xff

Receive (Acknowledge)	
Start	\xfe
Projector Address	\x01
Command [0]	\x00
Command [1]	\x06
Checksum	\x07
Stop	\xff

2.13 Orientation, Write

Description

Writes the orientation to the projector

Command

Command [0]	\x20
Command [1]	\x24

Data

Data [0] = unsigned byte

> **byte value:**
>
> **\x00 = rear/table ; \x40 = front/table ; \x80 = front/ceiling ; \xc0 = rear/ceiling**

Example

Set the orientation to front/table.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x24
Data[0]	\x40

Transmit	
Checksum	\x85
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.14 Orientation, Read

Description

Reads the orientation of the projector

Command

Command [0]	\x21
Command [1]	\x24

Returned Data

Data [0] = unsigned byte

byte value:

\x00 = rear/table ; \x40 = front/table ; \x80 = front/ceiling ; \xc0 = rear/ceiling

Example

Read the orientation being : \x40 = front/table.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x24
Checksum	\x46
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x24
Data[0]	\x40
Checksum	\x86
Stop	\xff

2.15 Shutter, Write

Description

Activates the shutter

Command

Command [0]	\x20
Command [1]	\x42

Data

Data [0] = unsigned byte

> **byte value:**
> **\x00 = CLOSE ; \x01 = OPEN**

Example

opens the shutter

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x42
Data[0]	\x01
Checksum	\x64
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01

Receive (acknowledge)	
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

2.16 Shutter, Read

Description

Reads the shutter's position

Command

Command [0]	`\x21`
Command [1]	`\x42`

Returned Data

Data [0] = unsigned byte

byte value:

`\x00` **= CLOSE ;** `\x01` **= OPEN**

Example

Read the shutter being `\x01` = open

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x42`
Checksum	`\x64`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (awnser)	
Start	`\xfe`
Projector address	`\x01`

Receive (awnser)	
Command[0]	\x21
Command[1]	\x42
Data[0]	\x01
Checksum	\x65
Stop	\xff

2.17 Freeze, Write

Description

Freezes the projected image

Command

Command [0]	\x20
Command [1]	\x23

Data

Data [0] = unsigned byte

byte value:

\x00 = OFF; \x01 = ON

Example

activate the Freeze function

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x23
Data [0]	\x00
Checksum	\x44
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.18 Freeze, Read

Description

Reads whether the projected image is frozen or not.

Command

Command [0]	\x21
Command [1]	\x23

Returned Data

Data [0] = unsigned byte

---

byte value:

\x00 = OFF ; \x01 = ON

Example

Reads the Freeze function, being \x00 = OFF

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x23
Checksum	\x45
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x23
Data[0]	\x00
Checksum	\x45
Stop	\xff

2.19 Logo ON/OFF

Description

Enables or disables the logo.

Command

Command [0]	\xf1
Command [1]	\x81

Data

Data [0] = unsigned byte

byte value:

`0` **= logo OFF;** `1` **= logo ON**

Example

Enables the logo

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\xf1
Command[1]	\x81
Data[0]	\x01
Checksum	\x74
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.20 Background, Write

Description

Sets a particular background

Command

Command [0]	\x20
Command [1]	\x7B

Data

Data [0] = unsigned byte

	byte value:
	\x00 = Logo ; \x01 = Blue ; \x02 = Black

Example

Sets the background to "Logo"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x7b
Checksum	\x9c
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.21 Background, Read

Description

Reads the actual background

Command

Command [0]	\x21
Command [1]	\x7b

Returned Data

Data [0] = unsigned byte

	byte value:
	\x00 = Logo ; \x01 = Blue ; \x02 = Black

Example

Reads the actual background, being \x00 = Logo

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x7b
Checksum	\x9d
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x7b
Data[0]	\x00
Checksum	\x9d
Stop	\xff

2.22 Aspect ratio, Write

Description

Sets the Aspect ratio

Command

Command [0]	\x20
Command [1]	\x0b

Data

Data [0] = unsigned byte

byte value:

\x00 = Auto ; \x01 = 4:3 ; \x02 = 5:4 ; \x03 = 16:9

Example

Sets the aspect ratio to "4x3"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x0b
Data[0]	\x01
Checksum	\x2d
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.23 Aspect Ratio, Read

Description

Reads the actual Aspect Ratio

Command

Command [0]	\x21
Command [1]	\x0b

Returned Data

Data [0] = unsigned byte

byte value:

0 = Auto ; 1 = 4:3 ; 2 = 5:4 ; 3 = 16:9

Example

Read the PiP window which has the focus, result = \x01 (4:3)

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0b
Checksum	\x2d
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0b
Data [0]	\x01
Checksum	\x2e
Stop	\xff

2.24 Show Native resolution (ON/OFF), Write

Description

Enable/disables the native resolution function

Command

Command [0]	\x20
Command [1]	\xaf

Sent Data

Data[0] : one byte : 0 = OFF ; 1 = ON

2.25 Show Native resolution (ON/OFF), Read

Description

Reads whether the native resolution function is enabled or not

Command

Command [0]	\x21
Command [1]	\xaf

Reaturned Data

Data[0] : one byte : 0 = OFF ; 1 = ON

2.26 Full screen representation(ON/OFF), Write

Description

Enable/disables the full screen representation

Command

Command [0]	\x20
Command [1]	\xae

Sent Data

Data[0] : one byte : 0 = OFF ; 1 = ON

2.27 Full screen representation (ON/OFF), Read

Description

Reads whether the Full screen representation is enabled or not

Command

Command [0]	\x21
Command [1]	\xae

Reaturned Data

Data[0] : one byte : 0 = OFF ; 1 = ON

2.28 Keystone, Write

Description

Adjusts the keystone

Command

Command [0]	\x20
Command [1]	\x50

Data

Signed word representing the keystone value.

Data[0] = msb

Data[1] = lsb

Example

Sets the keystone to "0"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x50
Data[0]	\x00

Transmit	
Data[1]	\x00
Checksum	\x71
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.29 Keystone, Read

Description

Reads the actual Keystone value

Command

Command [0]	\x21
Command [1]	\x50

Returned Data

signed word

Data[0] = MSB

Data[1] = LSB

Example

Reads the actual keystone value, being \x00

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x50
Checksum	\x72
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x50
Data[0]	\x00
Data[1]	\x00
Checksum	\x72
Stop	\xff

2.30 Lamp commands

Overview

- Lamp Mode, Write
- Lamp Mode, Read
- Lamp Status, Read
- Lamp Warning, Read
- Lamp runtime, Read
- Lamps Serial number, Read

2.30.1 Lamp Mode, Write

Description

Writes the Lamp mode to the projector

Command

Command [0]	\x20
Command [1]	\x0c

Data

Data[0]: unsigned byte

data byte value: 0 = Single mode ; 1= Dual mode

Example

Write the mode \x00 (single mode) to the projector

Transmit	
Start	\xfe
Projector address	\x01

Transmit	
Command[0]	\x20
Command[1]	\x0c
Data[0]	\x00
Checksum	\x2d
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.30.2 Lamp Mode, Read

Description

Reads the actual Lamp mode of the projector

Command

Command [0]	\x21
Command [1]	\x0c

Returned Data

Data[0]: unsigned byte

> **data byte value: 0 = Single mode ; 1= Dual mode**

Example

Reads the actual Lamp mode of the projector, the awnser being \x00 = single mode.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0c
Checksum	\x44
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01

Receive (acknowledge)	
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0c
Data [0]	\x00
Checksum	\x2e
Stop	\xff

2.30.3 Lamp Status, Read

Description

Reads the actual Lamp Status of the projector

Command

Command [0]	\x21
Command [1]	\x0d

Returned Data

Data [0] = Lamp Mode = unsigned byte see "Lamp Mode, Read", page 63

Data [1] = Active lamp(s) = unsigned byte (see following table)

Lamp	byte	Hex
Lamp1	0001	\x01
Lamp2	0010	\x02
Lamp1+Lamp2	0011	\x03

Table 2-209
Lamp status: active lamps table

Example

Reads the actual Lamp Status of the projector, the awnser being \x01 = Lamp1 (in single mode)

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0d
Checksum	\x2f
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0d
Data [0]	\x00
Data [1]	\x01
Checksum	\x30
Stop	\xff

2.30.4 Lamp Warning, Read

Description

Reads the actual Lamp Warning setting of the projector

Command

Command [0]	\x21
Command [1]	\x0e

Returned Data

Unsigned word composed of 2 bytes.see chapter "1. Serial communications Basics", "Data words", page 6

Data [0] = first byte: msb

Data [1] = second byte : lsb

Example

Reads the actual Lamp warning setting of the projector, the awnser being \x1E =30 hours

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0e
Checksum	\x30
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x0d
Data [0]	\x00
Data [1]	\x1e
Checksum	\x4d
Stop	\xff

2.30.5 Lamp runtime, Read

Description

Reads the lamp runtime, the lamp (lamp1 or lamp2) runtime to be read is given in the data field.

Command

Command [0]	\x64

Data

Data[0] = unsigned byte:

data value: 1 = lamp1 ; 2 = lamp2

Returned Data

Data[0] = unsigned byte (1 = lamp1 ; 2 = lamp2)

unsigned double word (4 bytes) representing the runitme : see chapter "1. Serial communications Basics", "Data words", page 6

Data[1] = msb

Data[2]

Data[3]

Data[4] = lsb

2.30.6 Lamps Serial number, Read

Description

Reads the Lamps Serial number, the lamp (lamp1 or lamp2) to be read is given in the data field.

Command

Command [0]	\x63

Data

Data[0] = unsigned byte

	data value: 1= lamp1 ; 2=lamp2

Returned Data

string = Pascal-language string.

2.31 Image files

2.31.1 File, write

Description

This command writes an image file to the projector, this can be used to write to the projector a previously backed up file (on PC) which has been read with the "file, read" command.

The string used to represent the filename is of the C-string type.

Command

Command [0]	\xbe

Data

strFilename = string of the C-string type representing the filename: Example of filename : **HD_1080P@60.xml**

	By default the file will be selected in the Custom directory. If the necessary file is a standard file then strFilename should be /Standard/HD_1080@60.xml.

FileData = data (ASCII coded) in XML format representing the file.

Example of FileData for the **HD_1080P@60.xml** file :

```xml
<?xml version="1.0" encoding="UTF-8" ?>
 <ImageFile>
    <FileSelection>
        <InputLayer />
        <SourceType />
        <LineDuration Discrepancy="220" Unit="ns">14840</LineDuration>
        <TotalLines Discrepancy="1">1118</TotalLines>
        <Interlaced>no</Interlaced>
    </FileSelection>
    <ImageDefinition>
        <TotalPixels>2536</TotalPixels>
        <ActivePixels>1920</ActivePixels>
        <PixelStart>380</PixelStart>
        <ActiveLines>1080</ActiveLines>
        <LineStart>31</LineStart>
        <AspectRatio>16:9</AspectRatio>
        <ColorSpaceFormat>ITU.BT-601</ColorSpaceFormat>
    </ImageDefinition>
 </ImageFile>
```

	The data must be XML "valid" i.e. all the rules (for example for opening/closing tags) must be respected.

> **The FileData should be the same as the data read using the "file, read" command.**

2.31.2 File, read

Description

This command reads an image file from the projector (for backup purposes, ...).

The string used to represent the filename is of the C-string type.

Command

Command [0]	\xbf

Send Data

strFilename = string of the C-string type representing the filename.

Returned Data

strFilename = string of the C-string type representing the filename.

FileData = binary data contained in the file.

2.31.3 File List, read

Description

This command reads a list of image files from the projector.

Wildcards can be used to specify the filenames.

Command

Command [0]	\xc0

Send Data

strFilemask1

...

strFilemaskN

Returned Data

strFilename1 = string of the C – string type representing the 1st filename.

...

strFilenameN = string of the C – string type representing the Nth filename.

2.31.4 Active files list, read

Description

This command reads a list of the active image files from the projector.

Command

Command [0]	\xc5

Returned Data

strFilename1 = string of the C – string type representing the active file in the Data window 1.

strFilename2 = string of the C – string type representing the active file in the Video window 1.

strFilename3 = string of the C – string type representing the active file in the Data window 2.

strFilename4 = string of the C – string type representing the active file in the Video window 2.

2.32 Color temperature

Overview

- • Color temperature (Red/Blue), Increment
- • Color temperature (Red/Blue), Decrement
- • Color temperature Red/Bleu, Write
- • Color temperature Red/Bleu, Read
- • Color temperature, Read

2.32.1 Color temperature (Red/Blue), Increment

Description

Increments the Red/Blue Color temperature by one.

Command

Increments the Red color temperature by one

Command [0]	\x22
Command [1]	\x43

Increments the Blue color temperature by one

Command [0]	\x22
Command [1]	\x44

Example

Increment Red color temperature of a projector with address \x01 by one.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x22
Command[1]	\x43
Checksum	\x66
Stop	\xff

2.32.2 Color temperature (Red/Blue), Decrement

Description

Decrements the Red/Blue Color temperature by one.

Command

Decrement the Red color temperature by one

Command [0]	\x23
Command [1]	\x43

Decrement the Blue color temperature by one

Command [0]	\x23
Command [1]	\x44

Example

Decrement the Red color temperature of a projector with address `\x01` by one.

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x23`
Command[1]	`\x43`
Checksum	`\x67`
Stop	`\xff`

2.32.3 Color temperature Red/Bleu, Write

Description

Writes a custom value for the different color temperatures to the projector

Commands

Red color temperature

Command [0]	`\x20`
Command [1]	`\x43`

Blue color temperature

Command [0]	`\x20`
Command [1]	`\x44`

Data

Unsigned word (2byte: msb-lsb).see chapter "1. Serial communications Basics", "Data words", page 6

Data [0] = unsigned byte(msb)

Data [1] = unsigned byte(lsb)

2.32.4 Color temperature Red/Bleu, Read

Description

Reads the color temperature setting for the different colortemperatures from the projector

Commands

reads Red color temperature

Command [0]	`\x21`
Command [1]	`\x43`

reads Blue color temperature

Command [0]	`\x21`
Command [1]	`\x44`

Send Data

Data[0] = unsigned byte representing the colortemperature to be read

Color temperature	Data[0]
Projector white	\x00
Broadcast	\x20
Film	\x36
Video	\x41
Computer	\x5d

If Data[0] is ommited current used colortemp is returned

Returned Data

Unsigned 2 byte word (msb-lsb).see chapter "1. Serial communications Basics", "Data words", page 6

Data [0] = unsigned byte (msb)

Data [1] = unsigned byte (lsb)

2.32.5 Color temperature, Read

Description

Reads the actual preset color temperature value from the projector

Commands

Command [0]	\x21
Command [1]	\x45

Returned Data

Data [0] = unsigned byte

data values: \x00 = projector white ; \x20 = broadcast ; \x36 = Film ;\x41 = Video ; \x5D = Computer ; \xff = custom

Example

Reads the actual preset color temperature of the projector, the awnser being \x00 = projector white

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x45
Checksum	\x67
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00

Receive (acknowledge)	
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x45
Data [0]	\x00
Checksum	\x67
Stop	\xff

2.33 Standby Timer, Write

Description

Adjusts the Standby Timer in a range from 180 to 3600.

Command

Command [0]	\x20
Command [1]	\x98

Data

Unsigned word representing the standby time value:

Data[0] = msb

Data[1] = lsb

> **range: 180 to 3600 seconds**
>
> **0 seconds = Standby timer OFF**

Example

Sets the standby time to "400"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x98
Data[0]	\x01
Data[1]	\x90
Checksum	\x4a
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.34 Standby Timer, Read

Description

Reads the actual Standby Timer value

Command

Command [0]	\x21
Command [1]	\x98

Returned Data

Unsigned word representing the standby time value:

Data[0] = msb

Data[1] = lsb

> **range: 180 to 3600 seconds**
>
> **0 seconds = Standby timer OFF**

Example

Reads the actual standby timer value, being \x00 (Timer OFF)

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x98
Checksum	\xba
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x98
Data[0]	\x00
Data[1]	\x00
Checksum	\xba
Stop	\xff

2.35 Switching mode, Write

Description

Writes the switching mode to the projector

Command

Command [0]	\x20
Command [1]	\x9a

Data

Data [0] = unsigned byte

Byte value	Switching mode
\x01	Seamless
\x02	Box in
\x03	Box out
\x04	shift in left
\x05	shift in right
\x06	shift in top
\x07	shift in bottom
\x08	vertical curtain open
\x09	vertical curtain close
\x0a	Horizontal curtain open
\x0b	Horizontal curtain close
\x0c	Fade in out

Example

Set the switching mode to seamless

Transmit	
Start	\xfe
Projector address	\x01

Transmit	
Command[0]	\x20
Command[1]	\x9a
Data[0]	\x01
Checksum	\xbc
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.36 Switching mode, Read

Description

Reads the Switching Mode of the projector

Command

Command [0]	\x21
Command [1]	\x9a

Returned Data

Data [0] = unsigned byte

Byte value	Switching mode
\x01	Seamless
\x02	Box in
\x03	Box out
\x04	shift in left
\x05	shift in right
\x06	shift in top
\x07	shift in bottom
\x08	vertical curtain open
\x09	vertical curtain close
\x0a	Horizontal curtain open
\x0b	Horizontal curtain close
\x0c	Fade in out

Example

Read the switching mode being `\x01` = seamless.

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x9a`
Checksum	`\xbc`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (Awnser)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\x9a`
Data[0]	`\x01`
Checksum	`\xbd`
Stop	`\xff`

2.37 Filmmode detection, Increment/Decrement

Description

Increments/decrements the Filmmode detection setting (on/off)

Commands

increment :

Command [0]	`\x22`
Command [1]	`\xa1`

decrement :

command[0]	`\x23`
command[1]	`\xa1`

2.38 Filmmode detection, Write

Description

Enables/disables the filmmode detection.

Command

Command [0]	\x20
Command [1]	\xa1

Data

Data [0] = byte type

byte value:

0 = OFF; 1 = ON

Example

Enables the Filmmode detection

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\xa1
Data[0]	\x01
Checksum	\xc3
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.39 Filmmode detection, Read

Description

Reads whether the Filmmode detection is enabled or not

Command

Command [0]	\x21
Command [1]	\xa1

Returned Data

Data [0] = unsigned byte

📄 **byte value:**

`\x00` **= OFF ;** `\x01` **= ON**

Example

Reads the Automatic startup, being `\x00` = OFF

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\xa1`
Checksum	`\xc3`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

Receive (awnser)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x21`
Command[1]	`\xa1`
Data[0]	`\x00`
Checksum	`\xc3`
Stop	`\xff`

2.40 Autoimage execute

Description

Executes the Autoimage function on the window defined by the first data byte, the second byte sets the Autoimage settings.

If the data bytes are ommited the Autoimage function will be executed on the active window (window which has the focus) using the actual Autoimage settings.

Command

Command [0]	\xf5

Data (optional)

Data[0] = byWindow

Data[1], Data[2] = wValue = word describing the settings of the AutoImage function

The 4 bits of the first byte (lsb) are used.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	Start pixel-line	Phase	Contr/Bright	TotalPixels

Table 2-264
byte description

Example

executes Autoimage on the window which has the focus (the data bytes are ommited).

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\xf5
Checksum	\xf6
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

2.41 Blanking

Overview

- Blanking Reset
- Blanking enable, Write
- Blanking enable, Read
- Blanking, Write
- Blanking, Read
- Blanking, Increment
- Blanking, Decrement
- Minimum Blanking, Read
- Maximum Blanking, Read

2.41.1   Blanking Reset

Description

Allows the reset of the blanking values.

Command

Command [0]	\x17

Data

byTypeBlanking : 1 byte describing the type of blanking.

Data [0] = 0 : blanking done and saved in the image file

Data [0] = 1: blanking done on the display

The maximum number of pixels which can be blanked depends on the native resolution of the panels

2.41.2   Blanking enable, Write

Description

Enables the blanking.

Command

Command [0]	\x20
Command [1]	\x56

Data

Data[0] = byTypeBlanking

Data[0] = 0 : blanking on image file

Data[0] = 1: blanking on display

Data[1] = unsigned byte

Data[1] = 0 : disable

Data[1] = 1 : enable

2.41.3   Blanking enable, Read

Description

Reads whether the blanking is enabled.

Command

Command [0]	\x21
Command [1]	\x56

Sent Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Returned Data

Data[0] = byTypeBlanking

Data[0] = 0 : blanking on image file

Data[0] = 1: blanking on display

Data[1] = unsigned byte

Data[1] = 0 : disable

Data[1] = 1 : enable

2.41.4 Blanking, Write

Description

Blanks the display.

Four blanking methods are possible:

- Top
- Bottom
- Left
- Right

The blanking may also be chosen to be performed on the display or on the image file (saved in the image file).

Command

Top blanking:

Command [0]	\x20
Command [1]	\x4c

Bottom blanking:

Command [0]	\x20
Command [1]	\x4d

Left blanking:

Command [0]	\x20
Command [1]	\x4e

Right blanking:

Command [0]	\x20
Command [1]	\x4f

Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Wvalue : Unsigned word (2 bytes) representing the number of pixels to be blanked.

Data [1] = msb

Data [2] = lsb

The maximum number of pixels which can be blanked depends on the native resolution of the panels

The panel size can be read using the Panel Size read command.
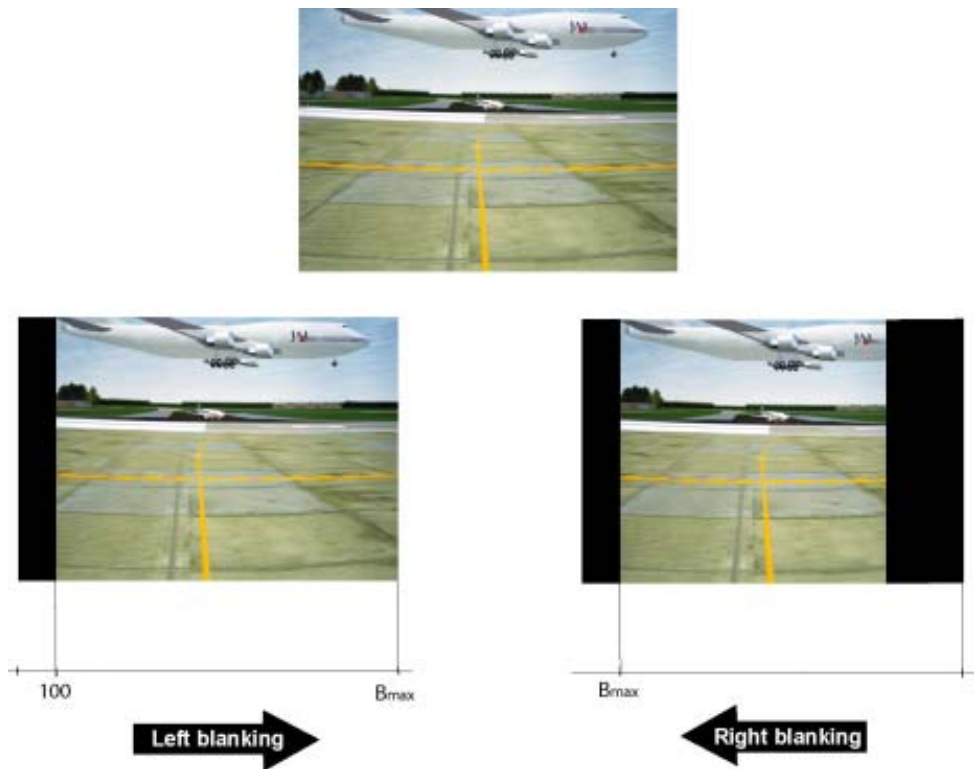
Image 2-1

2.41.5 Blanking, Read

Description

Reads the values of the blanking.

Command

Reads top blanking:

Command [0]	\x21
Command [1]	\x4c

Reads bottom blanking:

Command [0]	\x21
Command [1]	\x4d

Reads left blanking:

Command [0]	\x21
Command [1]	\x4e

Reads right blanking:

Command [0]	\x21
Command [1]	\x4f

Sent Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Returned Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Wvalue : Unsigned word (2 bytes) representing the number of pixels blanked.

Data [1] = msb

Data [2] = lsb

The maximum number of pixels blanked depends on the native resolution of the panels.
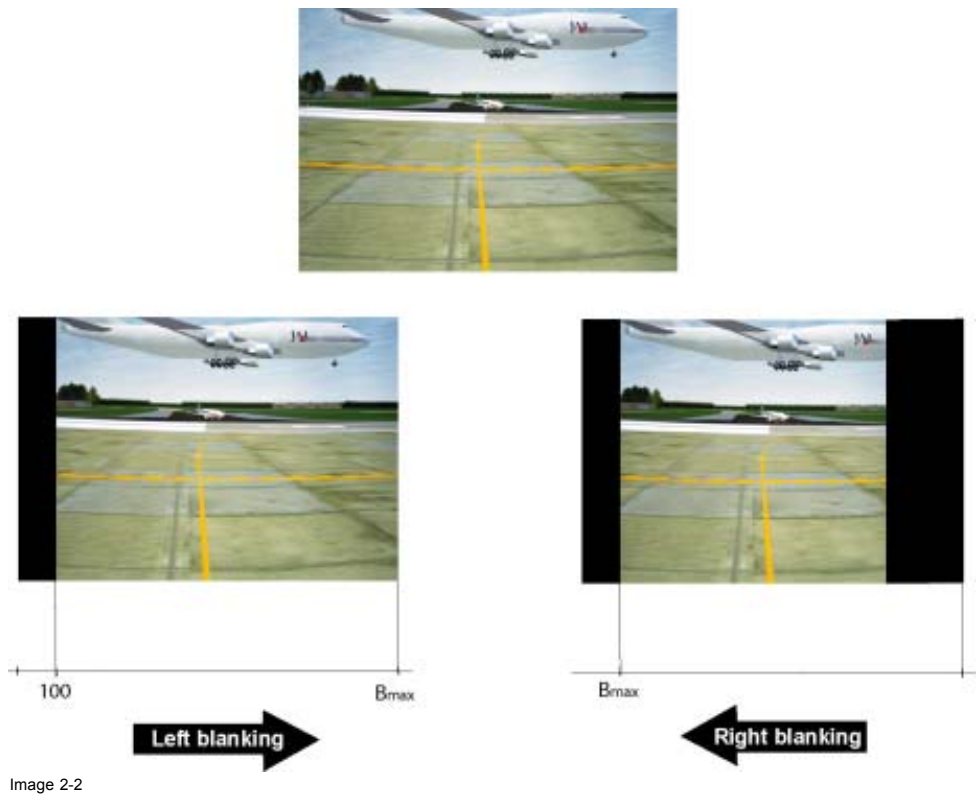
Image 2-2

2.41.6 Blanking, Increment

Description

Increments the blanking value by one.

Command

Increments Top blanking:

Command [0]	\x22
Command [1]	\x4c

Increments Bottom blanking:

Command [0]	\x22
Command [1]	\x4d

Increments Left blanking:

Command [0]	\x22
Command [1]	\x4e

Increments Right blanking:

Command [0]	\x22
Command [1]	\x4f

Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

WStep = unsigned word representing the amount of increments

The maximum number of pixels which can be blanked depends on the native resolution of the panels

2.41.7 Blanking, Decrement

Description

Decrements the blanking value by one.

Command

Decrements Top blanking:

Command [0]	\x23
Command [1]	\x4c

Decrements Bottom blanking:

Command [0]	\x23
Command [1]	\x4d

Decrements Left blanking:

Command [0]	\x23
Command [1]	\x4e

Decrements Right blanking:

Command [0]	\x23
Command [1]	\x4f

Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

WStep = unsigned word representing the amount of decrements

The maximum number of pixels which can be blanked depends on the native resolution of the panels

2.41.8 Minimum Blanking, Read

Description

Reads the Minimum values for the blanking.

Command

Reads minimal top blanking:

Command [0]	\x26
Command [1]	\x4c

Reads minimal bottom blanking:

Command [0]	\x26
Command [1]	\x4d

Reads minimal left blanking:

Command [0]	\x26
Command [1]	\x4e

Reads minimal right blanking:

Command [0]	\x26
Command [1]	\x4f

Sent Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Returned Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Unsigned word (2 bytes) representing the minimum value to be used for blanking.

Data [0] = msb

Data [1] = lsb

2.41.9 Maximum Blanking, Read

Description

Reads the Maximum values for the blanking.

Command

Reads minimal top blanking:

Command [0]	\x27
Command [1]	\x4c

Reads minimal bottom blanking:

Command [0]	\x27
Command [1]	\x4d

Reads minimal left blanking:

Command [0]	\x27
Command [1]	\x4e

Reads minimal right blanking:

Command [0]	\x27
Command [1]	\x4f

Sent Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Returned Data

byTypeBlanking : one byte describing the type of blanking

Data [0] = 0 : blanking on image file (saved in image file)

Data [0] = 1 : blanking on display

Unsigned word (2 bytes) representing the maximum value to be used for blanking.

Data [0] = msb

Data [1] = lsb

The maximum number of pixels which can be blanked depends on the native resolution of the panels

3. ADVANCED COMMANDS

Overview

- Lamp Commands
- Picture In Picture
- Viewport Active, Write
- Viewport Active, Read
- Viewport Active, Increment/decrement
- Viewport Active Min/Max, Read
- Viewport Area, Write
- Viewport Area, Read
- Viewport Area Min/Max, Read
- Desktop area, Write
- Desktop area, Read
- Digital Zoom
- Automatic startup, Write
- Automatic startup, Read
- Startup screen, Write
- Startup screen, Read
- Input balance
- AutoImage setup, Write
- AutoImage setup, Read
- Automatic Gain Control, write
- Automatic Gain Control, Read
- Manual Gain Control value, Write
- Manual Gain Control value, Read
- Gain Control, Increment/Decrement
- Minimum/Maximum AGC value, Read
- Automatic Gain Control interval, Write
- Automatic Gain Control Interval, Read
- Data input format, Increment/Decrement
- Data input format, write
- Data input format, Read
- Image files
- I²C diagnosis, Read
- Panel size, Read
- Logo status, Read
- Take screenshot (Logo)
- Save screenshot (Logo)
- Abort saving screenshot (Logo)
- Load logo

3.1 Lamp Commands

3.1.1 Reset Lamp runtime, Write

Description

Reset the Lamp runtime of the projector, the lamp(s) to reset are given in the Data field along with a Pascal string representing the serial number.

Command

Command [0]	\x68

Data

Data[0]: unsigned byte

---

 data byte value: 1= lamp1 ; 2= lamp2

Pascal type string of length 7 representing the serial number.

3.1.2 Lamp History, Read

Description

Reads the actual Lamp History of the projector

Command

Command [0]	\x21
Command [1]	\x0f

Returned Data

Array containing maximum 20 elements, each element containing a C– language strings and 5bytes:

• 1 byte for the lamp (1 = lamp1; 2 = lamp2)
• 1 C language string made of 6 bytes for the serial number and 1 STOP byte
• 4 bytes for the Runtime

	Data11	Data10	Data9	Data8	Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
1st el.	LAMP	SN	SN	SN	SN	SN	SN	STOP	RT	RT	RT	RT
...												
20th el	LAMP	SN	SN	SN	SN	SN	SN	STOP	RT	RT	RT	RT

3.2 Picture In Picture

Overview

- PiP Loop Focus
- PiP Focus, Read
- PiP Focus, Write
- PiP layout , Read
- PiP layout, Write
- PiP layout List, Read
- PiP current layout, Read
- PiP select layout, Write
- PiP save layout
- PiP save layout as
- PiP rename layout
- PiP remove Layout
- PiP remove window
- Addable sources List, read
- PiP add window
- PiP window size, Write
- PiP window size, Read
- PiP window size, increment/decrement
- PiP Window size, Read Min/Max
- PiP free window size, Write
- PiP free window size, Read
- PiP Window free size, Read Min/Max
- Window horizontal position, Write
- Window horizontal position, Read
- Window vertical position, Write
- Window vertical position, Read
- Window horizontal position, Increment/decrement
- PiP Window horizontal position, Read Min/Max
- PiP Window vertical position, Read Min/Max
- PiP Priority Write
- Priority, Read
- PiP List active sources, Read
- PiP last added window type, Read

3.2.1 PiP Loop Focus

Description

Activates the PiP Loop Focus mode on the Projector (allows to browse through the PiP windows)

Command

Command [0]	\x12
Command [1]	\x01

Example

Activate the PiP Loop Focus

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x01

Transmit	
Checksum	\x14
Stop	\xff

3.2.2 PiP Focus, Read

Description

Returns the PiP window which has the focus.

Command

Command [0]	\x12
Command [1]	\x02

Returned Data

Data [0] = Window type byte

Window type byte value:

0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B

Example

Read the PiP window which has the focus, result = \x01 (Data A)

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x02
Checksum	\x15
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x02
Data [0]	\x01

Receive (Awnser)	
Checksum	\x16
Stop	\xff

3.2.3 PiP Focus, Write

Description

Gives the focus to a particular PiP window .

Command

Command [0]	\x12
Command [1]	\x03

Data

Data [0] = Window type byte

	Window type byte value:
	0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B

Example

Gives the focus to a particular PiP window , here: \x01 (Data A)

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x03
Checksum	\x16
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.2.4 PiP layout , Read

Description

Returns the configuration of a particular layout.

Command

Command [0]	\x12
Command [1]	\x0f

Sent Data

strPipName = string representing the name of the layout

	The strPipname string is of the C-language type. see chapter "1. Serial communications Basics", "Syntax", page 9

Returned Data

PiPData = binary data describing the PiP layout.

The data used to write a layout using the "write layout"command should be of the same form.see "PiP layout, Write", page 92

The data in question is not meant to be analyzed but can be used to write to another projector for instance.

3.2.5 PiP layout, Write

Description

Configures a particular PiP layout.

Command

Command [0]	\x12
Command [1]	\x10

Sent Data

strPipName

	The strPipname string is of the C-language type. see chapter "1. Serial communications Basics", "Syntax", page 9

PiPData = binary data describing the PiP layout.

The data used to write a layout using the "write layout" command should be of the same form of the data read using the "read PIP layout" command.see "PiP layout , Read", page 91

The data to be send is the data previously read by the PiP layout read command.

3.2.6 PiP layout List, Read

Description

Returns the PiP layout list.

Number of layouts (N) may vary from 1 (factory layouts) to 10.

Command

Command [0]	\x12
Command [1]	\x04

Send data

One or more filter masks (wildcards) must be specified, the logic function used between the different filter masks is the OR function.

The mask is sent in the form of a string (thus always terminated by a the stop byte \x00).

strMask1

...

strMaskN

	An example of a wildcard is the asterix (*), the hexadecimal notation of the asterix being \x2a.

Returned Data

strPipname1

...

strPipnameN

	The strPipname and strMask strings are of the C-language type. see chapter "1. Serial communications Basics", "Syntax", page 9

Example

Reads the actual PiP layout list using the wildcard (*), i.e. returns all the layouts.

Result = Full screen, 2by2 layout.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x04
Data[0]	\x2a
Data[1]	\x00
Checksum	\x41
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x04
Data [0]	\x46
Data [1]	\x75
Data [2]	\x6c
Data [3]	\x6c
Data [4]	\x2d
Data [5]	\x73

Receive (Awnser)	
Data [6]	\x63
Data [7]	\x72
Data [8]	\x65
Data [9]	\x65
Data [10]	\x6e
Data [11]	\x00
Data [0]	\x32
Data [1]	\x62
Data [2]	\x76
Data [3]	\x32
Data [4]	\x20
Data [5]	\x72
Data [6]	\x61
Data [7]	\x73
Data [8]	\x74
Data [9]	\x65
Data [10]	\x72
Data [11]	\x00
Checksum	\x44
Stop	\xff

3.2.7   PiP current layout, Read

Description

Returns the current PiP layout.

Command

Command [0]	\x12
Command [1]	\x05

Returned Data

strPipname = string

> **The string is of the C-language type see chapter "1. Serial communications Basics", "Syntax", page 9**

Example

Reads the actual PiP layout , result = Full screen.

Transmit	
Start	\xfe
Projector address	\x01

Transmit	
Command[0]	\x12
Command[1]	\x05
Checksum	\x18
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x05
Data [0]	\x46
Data [1]	\x75
Data [2]	\x6c
Data [3]	\x6c
Data [4]	\x2d
Data [5]	\x73
Data [6]	\x63
Data [7]	\x72
Data [8]	\x65
Data [9]	\x65
Data [10]	\x6e
Data [11]	\x00
Checksum	\x58
Stop	\xff

3.2.8   PiP select layout, Write

Description

Sets a PiP layout.

Command

Command [0]	\x12
Command [1]	\x06

Data

strPipname = string

> **The string is of the C-language type see chapter "1. Serial communications Basics", "Syntax", page 9**

Example

Sets the PiP layout = Full screen.

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x06
Data[0]	\x46
Data[1]	\x75
Data[2]	\x6c
Data[3]	\x6c
Data[4]	\x2d
Data[5]	\x73
Data[6]	\x63
Data[7]	\x72
Data[8]	\x65
Data[9]	\x65
Data[10]	\x6e
Data[11]	\x00
Checksum	\x59
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.2.9 PiP save layout

Description

Saves the active PiP layout.

Command

Command [0]	\x12
Command [1]	\x07

3.2.10 PiP save layout as

Description

Saves a PiP layout .

Command

Command [0]	\x12
Command [1]	\x08

Data

strPipname = string of maximum 12 characters (including the \x00 byte)

> **The string is of the C-language type see chapter "1. Serial communications Basics", "Syntax", page 9**

Example

Saves the active PiP layout as "test".

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x08
Data[0]	\x74
Data[1]	\x65
Data[2]	\x73
Data[3]	\x74
Data[4]	\x00
Checksum	\xdb
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

3.2.11 PiP rename layout

Description

Renames a PiP layout .

Command

Command [0]	\x12
Command [1]	\x09

Data

strOldPipname = string (old name of the layout)

strNewPipname = string of max 12 characters (new layout name)

> **The string is of the C– language type see chapter "1. Serial communications Basics", "Syntax", page 9**

Example

Renames the PiP layout "test" to "test2".

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x09
Data[0]	\x74
Data[1]	\x65
Data[2]	\x73
Data[3]	\x74
Data[4]	\x00
Data[5]	\x74
Data[6]	\x65
Data[7]	\x73
Data[8]	\x74
Data[9]	\x32
Data[10]	\x00
Checksum	\xce
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.2.12 PiP remove Layout

Description

Removes a PiP layout.

Command

Command [0]	\x12
Command [1]	\x0a

Data

strPipname1

...

strPipnameN

> **The strPipnameN string is of the C-language type. see chapter "1. Serial communications Basics", "Syntax", page 9**
>
> **Wildcards can be used to specify strPipnameN.see "Syntax", page 9**

Example

Removes the PiP layout "test".

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x12
Command[1]	\x0a
Data[0]	\x74
Data[1]	\x65
Data[2]	\x73
Data[3]	\x74
Data[4]	\x00
Checksum	\xdd
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01

Receive (acknowledge)	
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

3.2.13 PiP remove window

Description

Removes a particular PiP window .

Command

Command [0]	`\x12`
Command [1]	`\x0b`

Data

Data [0] = Window type byte

> **Window type byte value:**
>
> **0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B**

Example

Removes a particular PiP window , here: `\x01` (Data A)

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x12`
Command[1]	`\x0b`
Data[0]	`\x01`
Checksum	`\x1f`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

3.2.14 Addable sources List, read

Description

This command reads a list of the sources which can be added to the PiP using the "add window" command.

Command

Command [0]	\x12
Command [1]	\x0d

Returned Data

strSourceName1

...

strFilenameN

3.2.15 PiP add window

Description

Adds a PiP window .

To add a window following have to be specified:

- source to be visualized in the window
- Mode of the source input
- position and size of the window
- order (priority) of the window (front, background)

Command

Command [0]	\x12
Command [1]	\x0c

Data

Data[0] = bySource, byte describing the source to be linked to the window.

Byte value	source
1	BNC's
2	D15
3	Composite video
4	SVideo
5	DVI
6	Firewire
7	SDI
8	PC

Data[1] = byMode, byte describing the signals connected to the input section.

For more info about the input configurations, see the projector's Owner's manual.

for BNC's and D15:

Byte value	Mode
0	RGBHV
1	RGBCS (analog)
2	RGBCS (digital)
3	RG_sB
4	PrYPb
255	detect slot mode

for composite video:

Byte value	Mode
0	Video BNC
1	Vertical sync BNC
2	Red BNC
3	Green BNC
4	Blue BNC
5	SVideo Y
6	SVideo C

for Svideo:

Byte value	Mode
100	SVideo 1
101	SVideo on R & B
102	SVideo on Vs and Video BNC

Data[2], Data[3] = Xsize word

Data[4], Data[5] = Ysize word

Data[6], Data[7] = Xposition word

Data[8], Data[9] = Yposition word

> **Xsize, Ysize, Xpos, Ypos are given in "pixels"**
>
> **for a graphics projector version : x = 1 ...1024 ; y = 1..768**
>
> **Note that the panel size (projector version) can be read using the Panel size command, "Panel size, Read ", page 159**

> **Xsize and Ysize determine the size of the window, since the aspect ratio is fixed, one of the values may be sufficient to determine the size of the window.**
>
> **If the 2 values do not match the aspect ratio, then Xsize is taken as reference and Ysize is calculated to match the aspect ratio.**

Data[10] = byPriority byte: 0 =Top ;...; 3= Bottom

3.2.16   PiP window size, Write

Description

Resizes the active window (works in PiP and in Full Screen mode) by keeping the same aspect ratio.
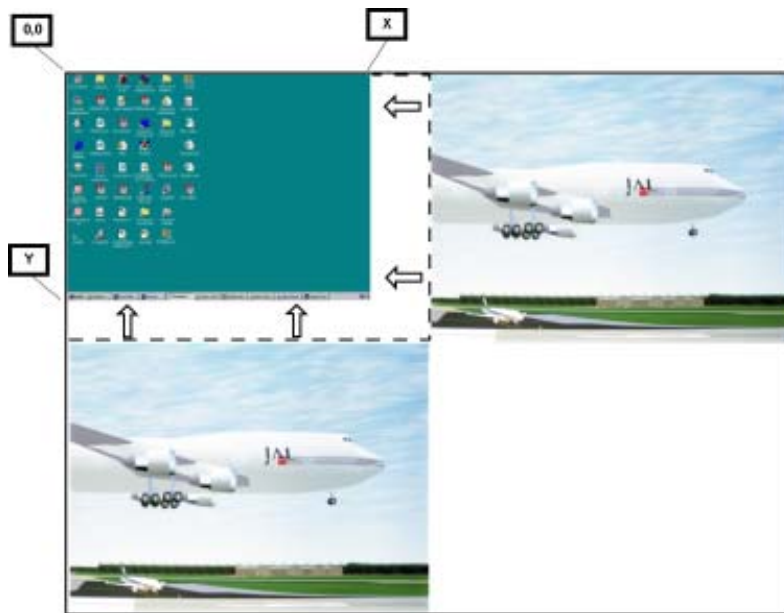
Image 3-1
PiP layout (3 windows, 2 video/1data), data window is resized

Command

Command [0]	\x20
Command [1]	\x91

Data

Data[0]...Data[3] = double word Xsize : describes the "width" of the window (x axis).

Data[4]...Data[7] = double word Ysize : describes the "height" of the window (y axis).

> **Xsize, Ysize are given in "pixels"**

> **Xsize and Ysize determine the size of the window, since the aspect ratio is fixed, one of the values may be sufficient to determine the size of the window.**
>
> **If the 2 values do not match the aspect ratio, then Xsize is taken as reference and Ysize is calculated to match the aspect ratio.**

3.2.17 PiP window size, Read

Description

Returns the size of the active window (works in PiP and in Full Screen mode)

Command

Command [0]	\x21
Command [1]	\x91

Returned Data

Data[0]...Data[3] = double word Xsize : describes the "width" of the window (x axis).

Data[4]...Data[7] = double word Ysize : describes the "height" of the window (y axis).

> **Xsize, Ysize are given in "pixels"**

> **Xsize and Ysize determine the size of the window, since the aspect ratio is fixed, one of the values may be sufficient to determine the size of the window.**
>
> **If the 2 values do not match the aspect ratio, then Xsize is taken as reference and Ysize is calculated to match the aspect ratio.**

3.2.18 PiP window size, increment/decrement

Description

Increments/decrements the size of the active wiindow (works in PiP and in Full Screen mode)

Command

increment

Command [0]	\x22
Command [1]	\x91

decrement

Command [0]	\x23
Command [1]	\x91

Data

Data[0] = byRepeat (optional)

> **Note that the panel size (projector version) can be read using the Panel size command,**

3.2.19 PiP Window size, Read Min/Max

Description

Reads the minimum/maximum value for the window size adjustments in PiP.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x91

read maximum:

Command[0]	\x27
Command[1]	\x91

Returned Data

Data[0]...Data[3] = signed double word Xsize : describes the "width" of the window (x axis).

Data[4]...Data[7] = signed double word Ysize : describes the "height" of the window (y axis).

> **Note that the panel size (projector version) can be read using the Panel size command**

3.2.20 PiP free window size, Write

Description

Allows the free resizing of the active window (works in PiP and in Full Screen mode) i.e the aspect ratio may be changed.
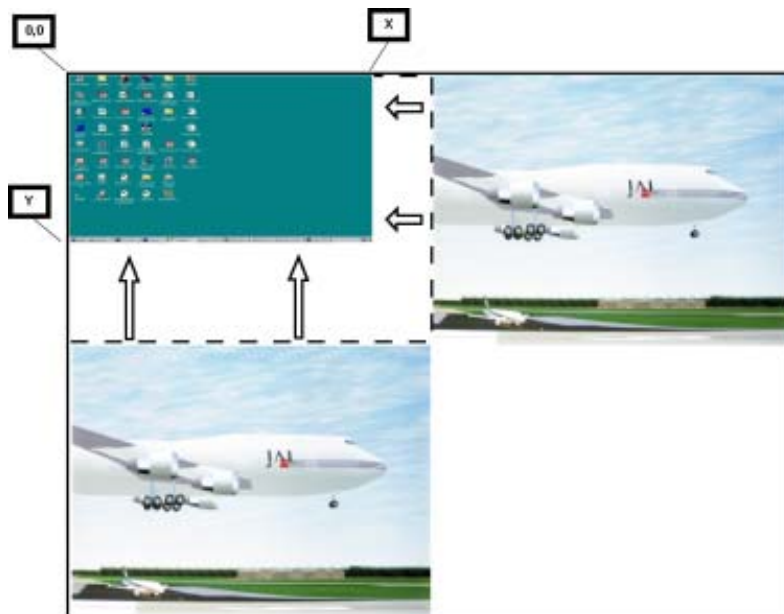
Image 3-2
PiP layout (3 windows, 2 video/1data), data window is resized

Command

Command [0]	\x20
Command [1]	\x92

Data

Data[0] = byFlag

bit0 = position : if bit0 is set to 1 the new position will be applied

bit1 = size: if bit1 is set to 1 the new size will be applied

bit2 = :return bit : if bit2 is set to 1 the new position and size will be returned by the projector.

bit3 = aspect ratio : if bit3 is set to 1 the aspect ratio will not be locked (free resizing)

Data[1]...Data[4] = signed double word XPosition : describes the start position of the window (x axis).

Data[5]...Data[8] = signed double word YPosition : describes the start position of the window (y axis).

Data[9]...Data[12] = signed double word Xsize : describes the "width" of the window (x axis).

Data[13]...Data[16] = signed double word Ysize : describes the "height" of the window (y axis).

> **If bit3 of byFlag is is set to 0 aspect ratio will not be altered, note that this command has then the same result as the (standard) PiP Window size command.**

> **Xsize, Ysize are given in "pixels"**

3.2.21 PiP free window size, Read

Description

Reads the size of the active window (works in PiP and in Full Screen mode)

Command

Command [0]	\x21
Command [1]	\x92

Returned Data

Data[0]...Data[3] = signed double word XPosition : describes the start position of the window (x axis).

Data[4]...Data[7] = signed double word YPosition : describes the start position of the window (y axis).

Data[8]...Data[11] = signed double word Xsize : describes the "width" of the window (x axis).

Data[12]...Data[15] = signed double word Ysize : describes the "height" of the window (y axis).

Xsize, Ysize are given in "pixels"

3.2.22 PiP Window free size, Read Min/Max

Description

Reads the minimum/maximum value for the window free size adjustments in PiP.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x92

read maximum:

Command[0]	\x27
Command[1]	\x92

Returned Data

Data[1]...Data[4] = signed double word XPosition : describes the min/max start position of the window (x axis).

Data[5]...Data[8] = signed double word YPosition : describes the min/max start position of the window (y axis).

Data[9]...Data[12] = signed double word Xsize : describes the min/max "width" of the window (x axis).

Data[13]...Data[16] = signed double word Ysize : describes the min/max "height" of the window (y axis).

Note that the panel size (projector version) can be read using the Panel size command

3.2.23 Window horizontal position, Write

Description

This command sets the horizontal position of the active window.
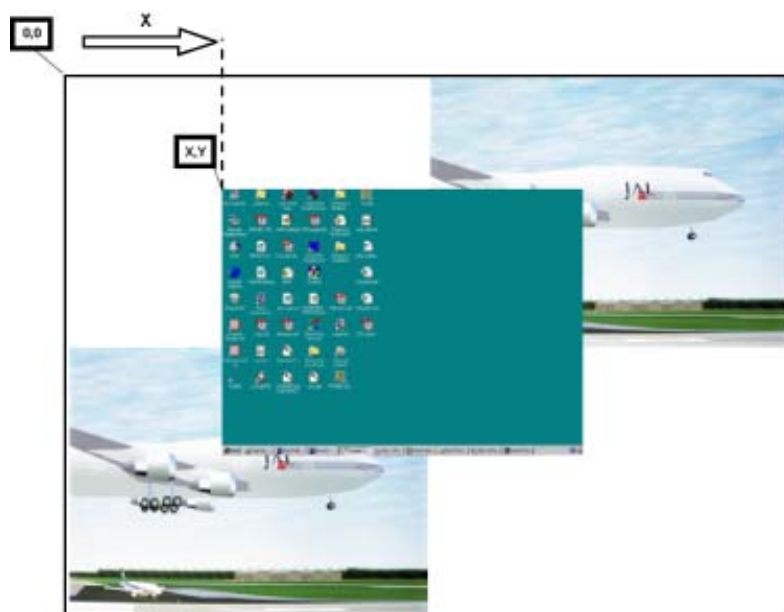
Image 3-3
PiP : data window is repositionned

Command

Command [0]	\x20
Command [1]	\x94

Data

Data [0]...Data[3] = signed double word

3.2.24 Window horizontal position, Read

Description

This command returns the horizontal position of the active window.

Command

Command [0]	\x21
Command [1]	\x94

Returned Data

Data [0]...Data[3] = signed double word

3.2.25 Window vertical position, Write

Description

This command sets the vertical position of the active window.
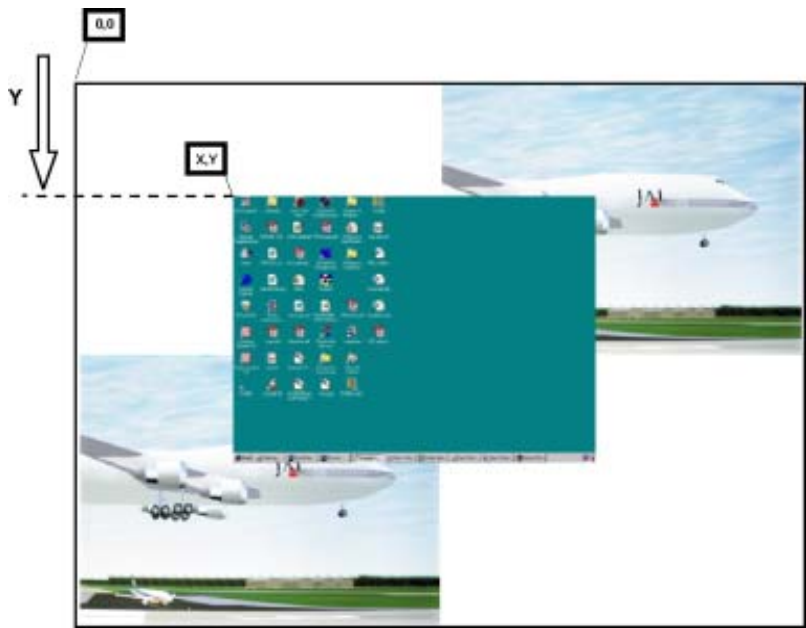
Image 3-4

Command

Command [0]	\x20
Command [1]	\x95

Data

Data [0]...Data[3] = signed double word

3.2.26   Window vertical position, Read

Description

This command returns the vertical position of the active window.

Command

Command [0]	\x21
Command [1]	\x95

Returned Data

Data [0]...Data[3] = signed double word

3.2.27   Window horizontal position, Increment/decrement

Description

This command increments/decrements the horizontal position of the active window.
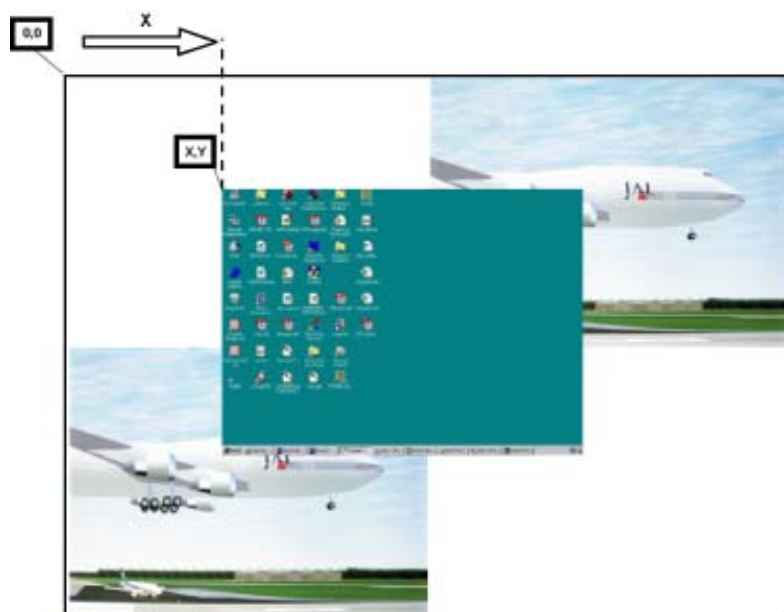
Image 3-5
PiP : data window is repositionned

Command

Increment

Command [0]	\x22
Command [1]	\x94

Decrement

Command [0]	\x23
Command [1]	\x94

Data

Data [0] = byRepeat : optional

3.2.28 PiP Window horizontal position, Read Min/Max

Description

Reads the minimum/maximum value for the window horizontal position adjustments in PiP.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x94

read maximum:

Command[0]	\x27
Command[1]	\x94

Returned Data

Data[0]...Data[3] = signed double word Xposition : describes the min/max "width" of the window (x axis).

	Note that the panel size (projector version) can be read using the Panel size command

3.2.29 PiP Window vertical position, Read Min/Max

Description

Reads the minimum/maximum value for the window vertical position adjustments in PiP.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x95

read maximum:

Command[0]	\x27
Command[1]	\x95

Returned Data

Data[0]...Data[3] = signed double word Yposition : describes the min/max "heigth" of the window (y axis).

	Note that the panel size (projector version) can be read using the Panel size command

3.2.30 PiP Priority Write

Description

Sets the priority (order) of a particular window

Command

Command [0]	\x12
Command [1]	\x0e
Command [2]	1,2,3

Command [2] : 1= Top ; 2 = bring to front ; 3 = send to back

Data

Data [0] = Window type byte

	Window type byte value:
	0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B

Example

Brings the Data A window to the front

Transmit	
Start	\xfe
Projector address	\x01

Transmit	
Command[0]	\x12
Command[1]	\x0e
Command[2]	\x02
Data[0]	\x00
Checksum	\x23
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.2.31 Priority, Read

Description

Reads the priority (order) of a particular window.

Command

Command [0]	\x12
Command [1]	\xoe
Command [2]	\x00

Data

Data [0] = Window type byte

> **Window type byte value:**
>
> **0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B**

Returned Data

Data [0] = unsigned byte

> **0 = Top ; ... ; 3 = bottom**

3.2.32 PiP List active sources, Read

Description

Returns the PiP active sources, for each window. The returned data is composed of 4 times a determined sequence.

Command

Command [0]	\x12
Command [1]	\x11

Returned Data

Returned data is 4 times following data :

Data [0] = Window type byte ; 0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B

Data [1] = Window visible byte ; 0 = NO ; 1 = YES

Data [2] = file found byte ; 0 = NO ; 1 = YES

Data [3] = input selection byte

for video windows : 0 = decoder layer1; 1 = decoder layer3 ; 4 = SDI ; 0xff = None

for data windows : 0 = data ; 1 = SBC ; oxff = None

Data [4] = source selection word

for video windows

word value	Source
0	video
1	VS
2	R
3	G
4	B
5	Lum
6	Chr
7	Video2
20	PrYPb on RGB
21	RGB video on RGB
22	RG_sB video on RGB
100	S-video
101	S-Video2
102	S-Video3
103	S-Video4
300	SDI
0xff	none

for data windows:

word value	source
0	RGBHV
1	$RGBC_S$ (digital)
2	$RGBC_S$ (analog)[6]
3	RG_SB
100	DssPryPb

---

6. the signal present on the Cs signal is a composite video signal containing the composite sync

word value	source
101	RGB Video
102	RG$_S$B Video
200	DVI
0xff	none

3.2.33 PiP last added window type, Read

Description

Returns the last added window type

Command

Command [0]	\x12
Command [1]	\x12

Returned Data

Data [0] = Window type byte ; 0 = Data A ; 1 = Video A ; 2 = Data B ; 3 = Video B

3.3 Viewport Active, Write

Description

Activates or disactivates the viewport function which allow to select an area of the image (window in PiP or full screen mode) to be visible.

Image 3-6

Command

Command [0]	\x20
Command [1]	\xB4

Data

Data[0] = one byte : 0 = not active ; 1 = active

3.4 Viewport Active, Read

Description

Returns the status of the viewport function which allow to select an area of the image (window in PiP or full screen mode) to be visible.

Command

Command [0]	\x21
Command [1]	\xB4

Returned Data

Data[0] = one byte : 0 = not active ; 1 = active

3.5 Viewport Active, Increment/decrement

Description

Increments/decrements the status of the viewport function which allow to select an area of the image (window in PiP or full screen mode) to be visible.

Command

increment

Command [0]	\x22
Command [1]	\xB4

decrement

Command [0]	\x23
Command [1]	\xB4

3.6 Viewport Active Min/Max, Read

Description

Returns the Min/Max value of the viewport function which allow to select an area of the image (window in PiP or full screen mode) to be visible.

Command

Minimum

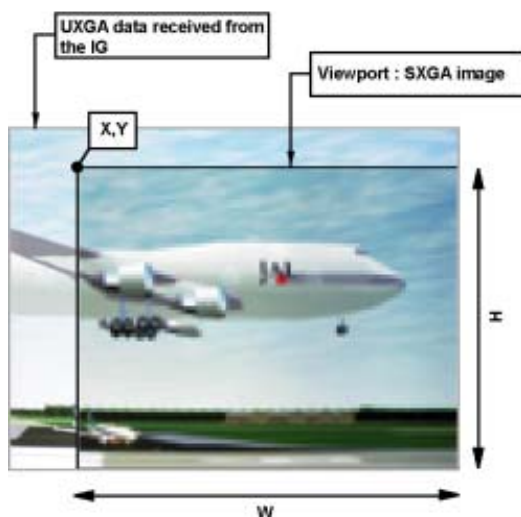| Command [0]	\x26
Command [1]	\xB4

Maximum

Command [0]	\x27
Command [1]	\xB4

Returned data

Data[0] = One byte describing the Min/Max value of the viewport active command.

3.7   Viewport Area, Write

Description

Sets the area for the viewport which allows to select an area of the image (window in PiP or full screen mode).
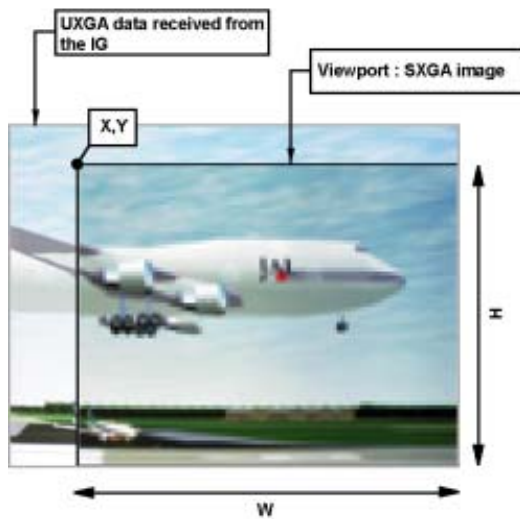
Image 3-7

Command

Command [0]	\x20
Command [1]	\xB5

Data

Data[0] = byFlag : mask byte

bit 0 = position : bit 0 = 0, position remains unchanged ; bit 0 = 1, position is adapted

Bit 1 = size : bit 1 = 0, size remains unchanged ; bit 1 = 1, size is adapted

Data[1 ...2] = wX : word (2 bytes) representing the X start position from the Left of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[3 ...4] = wY : word (2 bytes) representing the Y start position from the Top of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

Data[5 ...6] = wCX : word (2 bytes) representing the width of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[7 ...8] = wCY : word (2 bytes) representing the height of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

3.8   Viewport Area, Read

Description

Returns the viewport area.
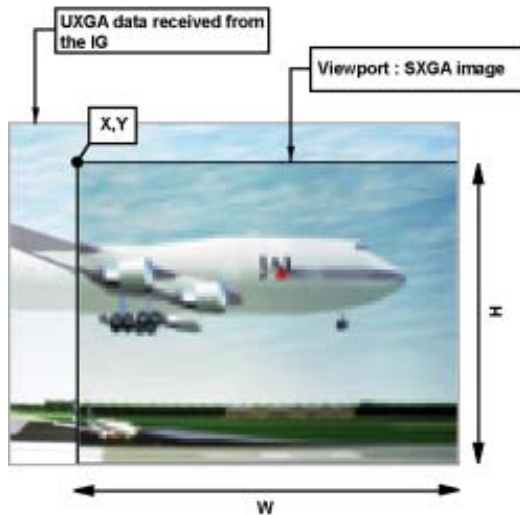
Image 3-8

Command

Command [0]	\x21
Command [1]	\xB5

Data

Data[0 ...1] = wX : word (2 bytes) representing the X start position from the Left of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[2 ...3] = wY : word (2 bytes) representing the Y start position from the Top of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

Data[4 ...5] = wCX : word (2 bytes) representing the width of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[6 ...7] = wCY : word (2 bytes) representing the height of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

3.9   Viewport Area Min/Max, Read

Description

Returns the minimum/maximum values for the viewport area.

Command

minimum

Command [0]	\x26
Command [1]	\xB5

maximum

Command [0]	\x27
Command [1]	\xB5

Data

Data[0 ...1] = wX : word (2 bytes) representing the X start position from the Left of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[2 ...3] = wY : word (2 bytes) representing the Y start position from the Top of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

Data[4 ...5] = wCX : word (2 bytes) representing the width of the viewport relative to the desktop (image). The value is a percentage of the desktop's width x 10000

Data[6 ...7] = wCY : word (2 bytes) representing the height of the viewport relative to the desktop (image). The value is a percentage of the desktop's height x 10000

3.10 Desktop area, Write

Description

Sets the area of the desktop.

For the display of a high resolution source one may use 2 projectors, the desktop area will set the portion of the image to be displayed by each projectors.
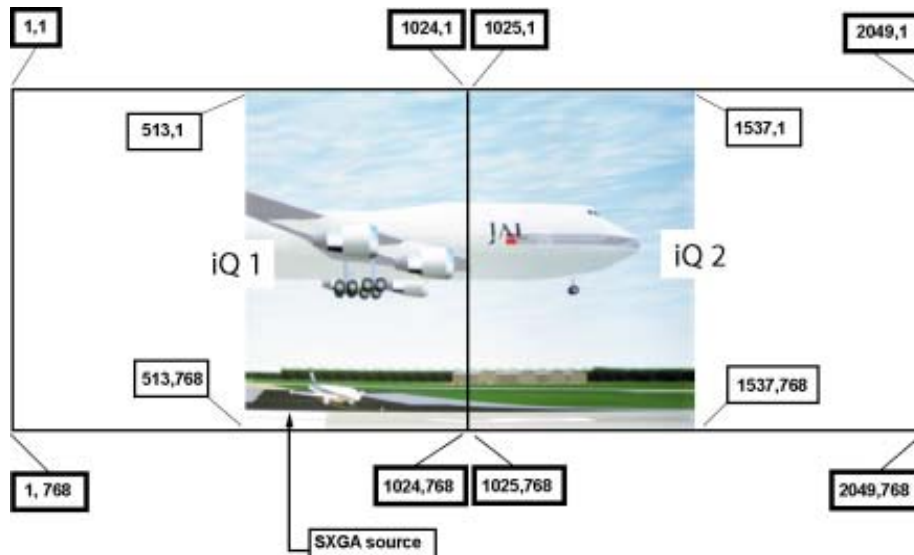
Image 3-9
Desktop setting

The positioning of the image is done with the Window position command

Command

Command [0]	\xEF
Command [1]	\x81

Data

Data[1]...Data[4] = double word Top : describes the Top of the desktop (example : 1).

Data[5]...Data[8] = double word Left : describes the Left of the desktop (example : 1).

Data[9]...Data[12] = double word Bottom : describes the Bottom of the desktop (example : 768).

Data[13]...Data[16] = double word Right : describes the Right of the desktop (example : 1024).

3.11 Desktop area, Read

Description

Returns the area of the desktop.
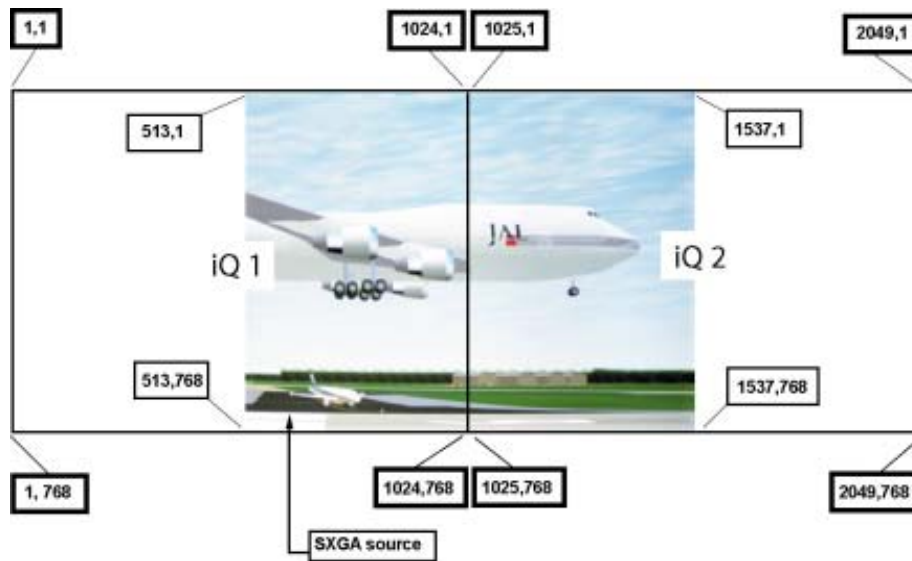
Image 3-10
Desktop setting

Command

Command [0]	\xEF
Command [1]	\x01

Data

Data[1]...Data[4] = double word Top : describes the Top of the desktop (example : 1).

Data[5]...Data[8] = double word Left : describes the Left of the desktop (example : 1).

Data[9]...Data[12] = double word Bottom : describes the Bottom of the desktop (example : 768).

Data[13]...Data[16] = double word Right : describes the Right of the desktop (example : 1024).

3.12 Digital Zoom

Overview

- Minimum/Maximum window size, Read
- Minimum/Maximum window X position, Read
- Minimum/Maximum window Yposition, Read
- Window size, Write
- Window X position (panning), Write
- Window Y position (panning), Write
- Window size, Read
- Window X position (panning), read
- Window Y position (panning), read
- Window size, Increment
- Window X position, Increment
- Window Y position, Increment
- Window size, Decrement
- Window X position, Decrement
- Window Y position, Decrement

3.12.1 Minimum/Maximum window size, Read

Description

Reads the minimum/maximum (range) zoom factor.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x81

read maximum:

Command[0]	\x27
Command[1]	\x81

Returned Data

Data [0]...Data[3] = double word (4 bytes) = min/max value

Data[4] = byte describing the decimal part (2 digits)

> **min size = 5%**
>
> **max size = 135.733 %**

3.12.2 Minimum/Maximum window X position, Read

Description

Reads the minimum/maximum (range) zoom window X position.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x82

read maximum:

Command[0]	\x27
Command[1]	\x82

Returned Data

Data [0]...Data [3] = double word (4 bytes) = min/max value

Data [4] = byte describing the decimal part (2 digits)

3.12.3 Minimum/Maximum window Yposition, Read

Description

Reads the minimum/maximum (range) zoom window Y position.

Commands

read minimum:

Command [0]	\x26
Command [1]	\x83

read maximum:

Command[0]	\x27
Command[1]	\x83

Returned Data

Data [0]...Data [3] = double word (4 bytes) = min/max value

data [4] = byte describing the decimal part (2 digits)

3.12.4 Window size, Write

Description

This command sets the size of the zoom window.

This is done by sending a percentage, 100% corresponds to the default size of the window.

Command

Command [0]	\x20
Command [1]	\x81

Data

Data [0]...Data[3] = double word (= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the desired percentage (100% = window size).

> 100 = zoom in

< 100 = zoom out

Data[4] = byte describing the decimal part (2 digits). This byte is Optional

3.12.5 Window X position (panning), Write

Description

This command sets the X position of the zoom window.

This is done by sending a percentage.

100% corresponds to a zoom window at the right border of the window.

0% corresponds to a zoom window at the left border of the window.

50% corresponds to a zoom window in the middle of the window.

Command

Command [0]	\x20
Command [1]	\x82

Data

Data [0]...Data[3] = double word(= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the desired percentage (100% = right border).

Data [4] = unsigned byte representing 2 decimal digits (0–100).This byte is optional

Values >100 are truncated to 100%

3.12.6 Window Y position (panning), Write

Description

This command sets the Y position of the active window (window which has the focus).

This is done by sending a percentage.

100% corresponds to a zoom window at the bottom of the window.

0% corresponds to a zoom window at the top of the window.

50% corresponds to a zoom window in the middle of the window.

Command

Command [0]	\x20
Command [1]	\x83

Data

Data [0]...Data [3] = double word(= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the desired percentage (100% = bottom).

Data [4] = unsigned byte representing 2 decimal digits (0–100). This byte is optional.

Values >100 are truncated to 100%.

3.12.7 Window size, Read

Description

This command reads the size of the zoom window.

Command

Command [0]	\x21
Command [1]	\x81

Returned Data

Data [0]...Data [3] = double word(= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the actual percentage (100% = initial window size).

> 100 = window has been zoomed in

< 100 = window has been zoomed out

Data[4] = unsigned byte : represents the decimal part

3.12.8 Window X position (panning), read

Description

This command reads the X position of the active window (window which has the focus).

100% corresponds to a zoom window at the right border of the window.

0% corresponds to a zoom window at the left border of the window.

50% corresponds to a zoom window in the middle of the window.

Command

Command [0]	\x21
Command [1]	\x82

Returned Data

Data [0]...Data [3] = double word(= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the actual X positon (100% = right border).

Data[4] = unsigned byte : represents the decimal part

3.12.9 Window Y position (panning), read

Description

This command reads the Y position of the active window (window which has the focus).

100% corresponds to a zoom window at the bottom of the window.

0% corresponds to a zoom window at the top of the window.

50% corresponds to a zoom window in the middle of the window.

Command

Command [0]	\x21
Command [1]	\x83

Data

Data [0]...Data [3] = double word(= 4 bytes; see chapter "1. Serial communications Basics", "Data words", page 6) corresponding to the actual Y position (100% = right border).

Data[4] = unsigned byte : represents the decimal part.

3.12.10 Window size, Increment

Description

This command increments the zoom factor of the active window (window which has the focus).

This is done by incrementing the size percentage, the rate of increment can also be specified.

Command

Command [0]	\x22
Command [1]	\x81

Data

Data [0] = byRepeat =1 byte to define the rate of increment

3.12.11 Window X position, Increment

Description

This command increments the X position of the zoom window.

This is done by incrementing the X position percentage, the rate of increment can also be specified

Command

Command [0]	\x22
Command [1]	\x82

Data

Data [0] = byRepeat =1 byte to define the rate of increment

3.12.12 Window Y position, Increment

Description

This command increments the Y position of the zoom window.

This is done by incrementing the Y position percentage, the rate of increment can also be specified

Command

Command [0]	\x22
Command [1]	\x83

Data

Data [0] = byRepeat =1 byte to define the rate of increment

3.12.13 Window size, Decrement

Description

This command decrements the size of the zoom window.

This is done by decrementing the size percentage, the rate of decrement can also be specified

Command

Command [0]	\x23
Command [1]	\x81

Data

Data [0] = byRepeat =1 byte to define the rate of decrement

3.12.14 Window X position, Decrement

Description

This command decrements the X position of the zoom window.

This is done by decrementing the X position percentage, the rate of decrement can also be specified

Command

Command [0]	\x23
Command [1]	\x82

Data

Data [0] = byRepeat =1 byte to define the rate of decrement

3.12.15 Window Y position, Decrement

Description

This command decrement the Y position of the zoom window.

This is done by decrementing the Y position percentage, the rate of decrement can also be specified

Command

Command [0]	\x23
Command [1]	\x83

Data

Data [0] = byRepeat =1 byte to define the rate of decrement

3.13 Automatic startup, Write

Description

Enables/disables the Automatic Startup.

Command

Command [0]	\x20
Command [1]	\x96

Data

Data [0] = byte type

> **byte value:**
>
> **0 = OFF; 1 = ON**

Example

Enables the Automatic startup

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x20`
Command[1]	`\x96`
Data[0]	`\x01`
Checksum	`\xb8`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

3.14 Automatic startup, Read

Description

Reads whether the Automatic startup is enabled or not

Command

Command [0]	`\x21`
Command [1]	`\x96`

Returned Data

Data [0] = unsigned byte

> **byte value:**
>
> **`\x00` = OFF ; `\x01` = ON**

Example

Reads the Automatic startup, being `\x00` = OFF

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x96
Checksum	\xb8
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x96
Data[0]	\x00
Checksum	\xb8
Stop	\xff

3.15 Startup screen, Write

Description

Enables/disables the Startup screen.

Command

Command [0]	\x20
Command [1]	\x99

Data

Data [0] = byte type

---

byte value:

0 = OFF; 1 = ON

Example

Enables the Startup screen

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x99
Data[0]	\x01
Checksum	\xbb
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.16 Startup screen, Read

Description

Reads whether the Startup screen is enabled or not

Command

Command [0]	\x21
Command [1]	\x99

Returned Data

Data [0] = unsigned byte

byte value:

\x00 = OFF ; \x01 = ON

Example

Reads the Automatic startup, being \x00 = OFF

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x99
Checksum	\xbb
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x99
Data[0]	\x00
Checksum	\xbb
Stop	\xff

3.17 Input balance

Overview

- Input black balance, Increment
- Input black balance, Decrement
- Input black balance, Write
- Input black balance, Read
- Input white balance, Increment
- Input white balance, Decrement
- Input white balance,Write
- Input white balance, Read

3.17.1 Input black balance, Increment

Description

Increments the black input balance for the different color channels

Commands

Black input balance Red

Command [0]	\x22
Command [1]	\x84

Black input balance Green

Command [0]	\x22
Command [1]	\x85

Black input balance Blue

Command [0]	\x22
Command [1]	\x86

3.17.2 Input black balance, Decrement

Description

Decrements the black input balance for the different color channels

Commands

Black input balance Red

Command [0]	\x23
Command [1]	\x84

Black input balance Green

Command [0]	\x23
Command [1]	\x85

Black input balance Blue

Command [0]	\x23
Command [1]	\x86

3.17.3 Input black balance, Write

Description

Writes the black input balance for the different color channels to the projector

Commands

Black input balance Red

Command [0]	\x20
Command [1]	\x84

Black input balance Green

Command [0]	\x20
Command [1]	\x85

Black input balance Blue

Command [0]	\x20
Command [1]	\x86

Data

Data [0] = lsb

Data [1] = msb

3.17.4 Input black balance, Read

Description

Reads the black input balance for the different color channels from the projector

Commands

Black input balance Red

Command [0]	\x21
Command [1]	\x84

Black input balance Green

Command [0]	\x21
Command [1]	\x85

Black input balance Blue

Command [0]	\x21
Command [1]	\x86

Read Data

Data [0] = lsb

Data [1] = msb

Example

Reads the black input balance for Red, being \x10

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x84
Checksum	\xa6
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x84
Data [0]	\x10
Data [1]	\x00

Receive (Awnser)	
Checksum	\xb6
Stop	\xff

3.17.5 Input white balance, Increment

Description

Increments the white input balance for the different color channels

Commands

White input balance Red

Command [0]	\x22
Command [1]	\x87

White input balance Green

Command [0]	\x22
Command [1]	\x88

White input balance Blue

Command [0]	\x22
Command [1]	\x89

3.17.6 Input white balance, Decrement

Description

Decrements the white input balance for the different color channels

Commands

White input balance Red

Command [0]	\x23
Command [1]	\x87

White input balance Green

Command [0]	\x23
Command [1]	\x88

White input balance Blue

Command [0]	\x23
Command [1]	\x89

3.17.7 Input white balance,Write

Description

Writes the white input balance for the different color channels to the projector

Commands

Black input balance Red

Command [0]	\x20
Command [1]	\x87

Black input balance Green

Command [0]	\x20
Command [1]	\x88

Black input balance Blue

Command [0]	\x20
Command [1]	\x89

Data

Data [0] = lsb

Data [1] = msb

3.17.8 Input white balance, Read

Description

Reads the White input balance for the different color channels from the projector

Commands

White input balance Red

Command [0]	\x21
Command [1]	\x87

White input balance Green

Command [0]	\x21
Command [1]	\x88

White input balance Blue

Command [0]	\x21
Command [1]	\x89

Returned Data

Data [0] = lsb

Data [1] = msb

Example

Reads the White input balance for Red, being \x10

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x86
Checksum	\xa8
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x86
Data [0]	\x10
Data [1]	\x00
Checksum	\xb8
Stop	\xff

3.18 AutoImage setup, Write

Description

Sets up the AutoImage i.e. selects the AutoImage parameters.

Command

Command [0]	\x20
Command [1]	\x90

Data

Unsigned word.see chapter "1. Serial communications Basics", "Data words", page 6

Data[0] = unsigned byte (msb)

Data[1] = unsigned byte (lsb)

Parameter	Data[0]	Data[1]	Hex Value
Total Pixels	0000	0001	\x01
Contrast/Brightness	0000	0010	\x02
Phase	0000	0100	\x04
Start pixel/line	0000	1000	\x08
Total pixels + cont/bright	0000	0011	\x03
...			

Example

Set up AutoImage for "Total Pixels"

Transmit	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x20`
Command[1]	`\x90`
Data[0]	`\x00`
Data[1]	`\x01`
Checksum	`\xb2`
Stop	`\xff`

Receive (acknowledge)	
Start	`\xfe`
Projector address	`\x01`
Command[0]	`\x00`
Command[1]	`\x06`
Checksum	`\x07`
Stop	`\xff`

3.19 AutoImage setup, Read

Description

Reads the AutoImage set up

Command

Command [0]	`\x21`
Command [1]	`\x90`

Returned Data

Unsigned word.see chapter "1. Serial communications Basics", "Data words", page 6

Data[0] = unsigned byte (msb)

Data[1] = unsigned byte (lsb)

Parameter	Data[0]	Data[1]	Hex Value
Total Pixels	0000	0001	`\x01`
Contrast/Brightness	0000	0010	`\x02`
Phase	0000	0100	`\x04`
Start pixel/line	0000	1000	`\x08`
Total pixels + cont/bright	0000	0011	`\x03`
...			

Example

reads the AutoImage set up, being :`\x01` = "Total Pixels"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x90
Checksum	\xB2
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x90
Data[0]	\x00
Data[1]	\x01
Checksum	\xb3
Stop	\xff

3.20 Automatic Gain Control, write

Description

Enables/disables the Automatic Gain Control.

Command

Command [0]	\x20
Command [1]	\x9b

Data

Data [0] = byte type

byte value:

0 = OFF; 1 = ON

Example

Enables the AGC

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x9b
Data[0]	\x01
Checksum	\xbd
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.21 Automatic Gain Control, Read

Description

Reads whether the AGC is enabled or not

Command

Command [0]	\x21
Command [1]	\x9b

Returned Data

Data [0] = unsigned byte

byte value:

\x00 = OFF ; \x01 = ON

Example

Reads the AGC setting, being \x00 = OFF

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x9b

Transmit	
Checksum	\xbd
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x9b
Data[0]	\x00
Checksum	\xbd
Stop	\xff

3.22 Manual Gain Control value, Write

Description

Sets the manual gain control value.

Command

Command [0]	\x20
Command [1]	\xa2

Data

Data [0] = lsb

Data [1] = msb

Example

Sets the AGC to 10

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\xa2
Data[0]	\x0a

Transmit	
Data[1]	\x00
Checksum	\xcd
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.23 Manual Gain Control value, Read

Description

Reads the actual gain control value

Command

Command [0]	\x21
Command [1]	\xa2

Returned Data

Data [0] = lsb

Data [1] = msb

Example

Reads the AGC value, being \x0A = 10

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa2
Checksum	\xc4
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa2
Data[0]	\x0a
Data[1]	\x00
Checksum	\xce
Stop	\xff

3.24 Gain Control, Increment/Decrement

Description

Increments/decrements the Gain Control value by one

Commands

increment:

Command [0]	\x22
Command [1]	\xa2

decrement:

command[0]	\x23
command[1]	\xa2

3.25 Minimum/Maximum AGC value, Read

Description

Reads the minimum/maximum AGC(Automatic Gain Control) value.

Commands

read minimum:

Command [0]	\x26
Command [1]	\xa2

read maximum:

Command[0]	\x27
Command [1]	\xa2

Returned Data

Data [0] = min/max value

3.26 Automatic Gain Control interval, Write

Description

Sets the AGC to short or long.

Command

Command [0]	\x20
Command [1]	\x9c

Data

Data [0] = byte type

	byte value:
	0 = short ; 1 = long

Example

Sets the AGC interval to "short"

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\x9c
Data[0]	\x00
Checksum	\xbd
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.27 Automatic Gain Control Interval, Read

Description

Reads the AGC interval (short or long)

Command

Command [0]	\x21
Command [1]	\x9c

Returned Data

Data [0] = unsigned byte

> **byte value:**
> **\x00 = short ; \x01 = long**

Example

Reads the Automatic startup, being \x00 = short

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x9c
Checksum	\xbe
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\x9c
Data[0]	\x00
Checksum	\xbe
Stop	\xff

3.28 Data input format, Increment/Decrement

Description

Increments/decrements the Data input format setting.

Commands

increment:

Command [0]	\x22
Command [1]	\xa4

decrement:

command[0]	\x23
command[1]	\xa4

3.29 Data input format, write

Description

Sets the data input format.

Command

Command [0]	\x20
Command [1]	\xa4

Data

Data [0] = byte type

Data format	byte value
RGB	0
CrYCb	1
HDTV(CrYCb)	2

3.30 Data input format, Read

Description

Reads the actual data input format setting.

Command

Command [0]	\x21
Command [1]	\xa4

Returned Data

Data [0] = byte type

Data format	byte value
RGB	0
CrYCb	1
HDTV(CrYCb)	2

3.31 Image files

Overview

- Delete File(s)
- Copy File
- Rename File
- Move File
- Image files settings, Write
- Image files settings, read
- Image files settings,Increment
- Image files settings, Decrement
- Read Minimum/Maximum Image files settings
- VCO, Increment
- VCO, decrement
- VCO,write
- VCO, Read
- Minimum VCO, Read
- Maximum VCO, Read
- Charge pump current, Increment/Decrement
- charge pump current, Write
- charge pump current , Read
- Minimum/Maximum charge pump current, Read
- Clamp position, Increment/Decrement
- Clamp position, Write
- Clamp position, Read
- Minimum/Maximum clamp position, Read
- Comb filter, Increment/Decrement
- comb filter, Write
- Comb filter, Read
- Minimum/Maximum comb filter, Read
- Horizontal sync polarity, Increment/Decrement
- Minimum/Maximum Horizontal sync polarity, Read
- Horizontal sync polarity, Read
- Horizontal sync polarity, Write
- Sync locking, Increment/Decrement
- Minimum/Maximum Horizontal Sync locking, Read
- Sync locking, Read
- Sync locking, Write

3.31.1 Delete File(s)

Description

This command deletes a list of image files from the projector.

Wildcards can be used to specify the filenames.see chapter "1. Serial communications Basics", "Syntax", page 9

Command

Command [0]	\xc1

Send Data

strFilename1

...

strFilenameN

3.31.2 Copy File

Description

This command copies an image file (source file) to another file (destination file).

	The destination file must always be a custom file."Syntax", page 9

Command

Command [0]	\xc2

Send Data

strFilenameFrom

strFilenameTo

	strFilenameFrom and strFilenameTo are strings of the C-string type."Syntax", page 9

3.31.3 Rename File

Description

This command renames an image file (source file) to another file (destination file).

	The source file and the destination file must always be custom files.see "Syntax", page 9

Command

Command [0]	\xc3

Send Data

strFilenameFrom

strFilenameTo

	strFilenameFrom and strFilenameTo are strings of the C-string type.see "Syntax", page 9

3.31.4 Move File

Description

This command moves an image file (source file) to another file (destination file).

When moving a file from a location, this location is not deleted and can be used to store another new file.

	The source file and the destination file must always be custom files."Syntax", page 9

Command

Command [0]	\xc4

Send Data

strFilenameFrom

strFilenameTo

	strFilenameFrom and strFilenameTo are strings of the C-string type."Syntax", page 9

3.31.5 Image files settings, Write

Description

These commands set the image files characterisitics

Commands

write total lines:

Command [0]	\x20
Command [1]	\x58

write active lines

Command [0]	\x20
Command [1]	\x59

write line start

Command [0]	\x20
Command [1]	\x5a

write period

Command [0]	\x20
Command [1]	\x5b

write total pixels

Command [0]	\x20
Command [1]	\x5c

write active pixels

Command [0]	\x20
Command [1]	\x5d

write pixel start

Command [0]	\x20
Command [1]	\x5e

write clamp width

Command [0]	\x20
Command [1]	\x68

Data

Commands	Data type
total lines	unsigned word (2 bytes)
active lines	unsigned word (2 bytes)
line start	unsigned word (2 bytes)
period	unsigned word (2 bytes)
total pixels	unsigned word (2 bytes)
active pixels	unsigned word (2 bytes)
pixel start	unsigned word (2 bytes)
clamp width	unsigned byte

3.31.6 Image files settings, read

Description

These commands reads the image files characterisitics

Commands

Read total lines:

Command [0]	\x21
Command [1]	\x58

Read active lines

Command [0]	\x21
Command [1]	\x59

Read line start

Command [0]	\x21
Command [1]	\x5a

Read period

Command [0]	\x21
Command [1]	\x5b

Read total pixels

Command [0]	\x21
Command [1]	\x5c

Read active pixels

Command [0]	\x21
Command [1]	\x5d

Read pixel start

Command [0]	\x21
Command [1]	\x5e

Read clamp width

Command [0]	\x21
Command [1]	\x68

Returned Data

Commands	Data type
total lines	unsigned word (2 bytes)
active lines	unsigned word (2 bytes)
line start	unsigned word (2 bytes)
period	unsigned word (2 bytes)
total pixels	unsigned word (2 bytes)
active pixels	unsigned word (2 bytes)
pixel start	unsigned word (2 bytes)
clamp width	unsigned byte

3.31.7 Image files settings,Increment

Description

These commands set the image files characterisitics

Commands

Increment total lines:

Command [0]	\x22
Command [1]	\x58

Increment active lines

Command [0]	\x22
Command [1]	\x59

Increment line start

Command [0]	\x22
Command [1]	\x5a

Increment period

Command [0]	\x22
Command [1]	\x5b

Increment total pixels

Command [0]	\x22
Command [1]	\x5c

Increment active pixels

Command [0]	\x22
Command [1]	\x5d

Increment pixel start

Command [0]	\x22
Command [1]	\x5e

Increment clamp width

Command [0]	\x22
Command [1]	\x68

3.31.8 Image files settings, Decrement

Description

These commands set the image files characterisitics

Commands

decrement total lines:

Command [0]	\x23
Command [1]	\x58

decrement active lines

Command [0]	\x23
Command [1]	\x59

decrement line start

Command [0]	\x23
Command [1]	\x5a

decrement period

Command [0]	\x23
Command [1]	\x5b

decrement total pixels

Command [0]	\x23
Command [1]	\x5c

decrement active pixels

Command [0]	\x23
Command [1]	\x5d

decrement pixel start

Command [0]	\x23
Command [1]	\x5e

decrement clamp width

Command [0]	\x23
Command [1]	\x68

3.31.9   Read Minimum/Maximum Image files settings

Description

These commands read the min/max values of the image files settings

Commands

minimum total lines:

Command [0]	\x26
Command [1]	\x58

maximum total lines

Command [0]	\x27
Command [1]	\x58

minimum active lines

Command [0]	\x26
Command [1]	\x59

maximum active lines

Command [0]	\x27
Command [1]	\x59

minimum line start

Command[0]	\x26
Command[1]	\x5a

maximum line start

Command[0]	\x27
Command[1]	\x5a

Minimum period

Command [0]	\x26
Command [1]	\x5b

maximum period

Command [0]	\x27
Command [1]	\x5b

minimum total pixels

Command [0]	\x26
Command [1]	\x5c

maximum total pixels

Command [0]	\x27
Command [1]	\x5c

minimum active pixels

Command [0]	\x26
Command [1]	\x5d

maximum active pixels

Command [0]	\x27
Command [1]	\x5d

minimum pixel start

Command [0]	\x26
Command [1]	\x5e

maximum pixel start

Command [0]	\x27
Command [1]	\x5e

minimum clamp width

Command [0]	\x26
Command [1]	\x68

maximum clamp width

Command [0]	\x27
Command [1]	\x68

3.31.10 VCO, Increment

> **VCO**
>
> The Voltage Controlled Oscillator is used in a Phase Locked Loop which allows to synchronise the data pocessing with the incoming source data.

Description

Increments the VCO by one

Command

Command [0]	\x22
Command [1]	\xa8

3.31.11 VCO, decrement

> **VCO**
>
> The Voltage Controlled Oscillator is used in a Phase Locked Loop which allows to synchronise the data pocessing with the incoming source data.

Description

Decrements the VCO by one.

Command

Command [0]	\x23
Command [1]	\xa8

3.31.12 VCO,write

Description

Writes a new VCO value to the projector

Command

Command [0]	\x20
Command [1]	\xa8

Data

Data [0] = VCO value

> **Data[0] = unsigned byte**
>
> **The range can be found with the "read max/min" commands:**
>
> **range min: "Minimum VCO, Read", page 151**
>
> **range max :"Maximum VCO, Read", page 151**

3.31.13 VCO, Read

Description

Reads the actual VCO value of the projector

Command

Command [0]	\x21
Command [1]	\xa8

Returned Data

Data [0] = VCO value

> **data type = unsigned byte**

Example

Reads the actual VCO value , the awnser being \x00

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa8
Checksum	\xca
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (Answer)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa7
Data [0]	\x00
Checksum	\xc9
Stop	\xff

3.31.14 Minimum VCO, Read

Description

Reads the minimum (range) VCO value of the projector

Command

Command [0]	\x26
Command [1]	\xa8

Returned Data

Data [0] = minimum VCO value

3.31.15 Maximum VCO, Read

Description

Reads the Maximum (range) VCO value of the projector

Command

Command [0]	\x27
Command [1]	\xa8

Returned Data

Data [0] = Maximum VCO value

3.31.16 Charge pump current, Increment/Decrement

Description

Increments/decrements the charge pump current by one

Commands

increment:

Command [0]	\x22
Command [1]	\x9d

decrement:

command[0]	\x23
command[1]	\x9d

3.31.17 charge pump current, Write

Description

Sets the charge pump current for the VCO

Command

Command [0]	\x20
Command [1]	\x9d

Data

Data [0] = byte type

3.31.18 charge pump current , Read

Description

Reads the charge pump current

Command

Command [0]	\x21
Command [1]	\x9d

Returned Data

Data [0] = unsigned byte

3.31.19 Minimum/Maximum charge pump current, Read

Description

Reads the minimum/max charge pump current for the VCO

Commands

read minimum:

Command [0]	\x26
Command [1]	\x9d

read maximum:

Command[0]	\x27
Command[1]	\x9d

Returned Data

Data [0] = minimum VCO value

3.31.20 Clamp position, Increment/Decrement

Description

Increments/decrements the Clamp position by one

Commands

increment:

Command [0]	\x22
Command [1]	\x9e

decrement:

command[0]	\x23
command[1]	\x9e

3.31.21 Clamp position, Write

Description

Sets the clamp position.

Command

Command [0]	\x20
Command [1]	\x9e

Data

Data [0] = byte type

3.31.22 Clamp position, Read

Description

Reads the clamp position

Command

Command [0]	\x21
Command [1]	\x9e

Returned Data

Data [0] = unsigned byte

3.31.23 Minimum/Maximum clamp position, Read

Description

Reads the minimum/max clamp position

Commands

read minimum:

Command [0]	\x26
Command [1]	\x9e

read maximum:

Command[0]	\x27
Command[1]	\x9e

Returned Data

Data [0] = min/max value

3.31.24 Comb filter, Increment/Decrement

Description

Increments/decrements the Comb filter setting (on/off) by one

Commands

increment:

Command [0]	\x22
Command [1]	\xa0

decrement:

command[0]	\x23
command[1]	\xa0

3.31.25 comb filter, Write

Description

Enables/disables the comb filter

Command

Command [0]	\x20
Command [1]	\xa0

Data

Data [0] = byte type

byte value:

0 = OFF; 1 = ON

Example

Enables the comb filter

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x20
Command[1]	\xa0
Data[0]	\x00
Checksum	\xc1
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

3.31.26 Comb filter, Read

Description

Reads whether the comb filter is enabled or not

Command

Command [0]	\x21
Command [1]	\xa0

Returned Data

Data [0] = unsigned byte

byte value:

\x00 = OFF ; \x01 = ON

Example

Reads the comb filter, being \x00 = OFF

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa0
Checksum	\xc2
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06
Checksum	\x07
Stop	\xff

Receive (awnser)	
Start	\xfe
Projector address	\x01
Command[0]	\x21
Command[1]	\xa0
Data[0]	\x00
Checksum	\xc2
Stop	\xff

3.31.27 Minimum/Maximum comb filter, Read

Description

Reads the minimum/max comb filter

Commands

read minimum:

Command [0]	\x26
Command [1]	\xa0

read maximum:

Command[0]	\x27
Command[1]	\xa0

Returned Data

Data [0] = min/max value

3.31.28 Horizontal sync polarity, Increment/Decrement

Description

Increments/decrements the horizontal sync polarity by one

Commands

increment:

Command [0]	\x22
Command [1]	\xa9

decrement:

command[0]	\x23
command[1]	\xa9

3.31.29 Minimum/Maximum Horizontal sync polarity, Read

Description

Reads the minimum/max Horizontal sync polarity

Commands

read minimum:

Command [0]	\x26
Command [1]	\xa9

read maximum:

Command[0]	\x27
Command[1]	\xa9

Returned Data

Data [0] = min/max value

3.31.30 Horizontal sync polarity, Read

Description

Reads the Horizontal sync polarity setting

Command

Command [0]	\x21
Command [1]	\xa9

Returned Data

Data [0] = byte : 0 = active low ; 1 = active high ; 2 = auto (determined by ADC)

3.31.31 Horizontal sync polarity, Write

Description

Sets the Horizontal sync polarity setting in the image file.

Command

Command [0]	\x20
Command [1]	\xa9

Data

Data [0] = byte : 0 = active low ; 1 = active high ; 2 = auto (determined by ADC)

3.31.32 Sync locking, Increment/Decrement

Description

Increments/decrements the Sync locking setting by one

Commands

increment:

Command [0]	\x22
Command [1]	\xab

decrement:

command[0]	\x23
command[1]	\xab

3.31.33 Minimum/Maximum Horizontal Sync locking, Read

Description

Reads the minimum/max Sync locking

Commands

read minimum:

Command [0]	\x26
Command [1]	\xab

read maximum:

Command[0]	\x27
Command[1]	\xab

Returned Data

Data [0] = byte : 0 = slow ; 1 = fast

3.31.34 Sync locking, Read

Description

Reads the Sync locking setting

Command

Command [0]	\x21
Command [1]	\xab

Returned Data

Data [0] = byte : 0 = slow ; 1 = fast

3.31.35 Sync locking, Write

Description

Sets the Sync locking (fast or slow)

Command

Command [0]	\x20
Command [1]	\xab

Data

Data [0] = byte : 0 = slow ; 1 = fast

3.32 I²C diagnosis, Read

Description

Reads the I²C diagnosis.

Command

Command [0]	\x81
Command [1]	\x01

Sent Data

Data[0] = bySlaveAddr1

Data[1] = bySlaveAddr2

...

Data[N-1] = bySlaveAddrN

---

bySlaveAddrn = byte representing the address of the module or unit to be diagnosed.

If ommited, all all the units are reported.

Unit	Hex address
Motor driver	`\x74`
Lamp driver	`\x78`
Video processor	`\x96`
Fpga backplane1	`\xb4`
Fpga backplane2	`\xb6`
Fpga backplane3	`\xb8`
ADC 9886 layer1	`\x98`
ADC 9887 layer1	`\x9e`
ADC 9887 layer2	`\x9a`
DVI indicators layer1	`\x7e`
DVI indicators layer2	`\x7a`
Video decoder	`\x42`
Video indicators	`\x7c`

Table 3-286
address overview

Returned Data

Data[0] = strDiagnosis1

Data[1] = strDiagnosis2

...

Data[N-1] = strDiagnosisN

3.33 Panel size, Read

Description

Reads the size of the LCD panels

Command

Command [0]	`\xf0`
Command [1]	`\x01`

Returned Data

Data [0]...Data [3] = wpixels = unsigned word,see chapter "1. Serial communications Basics", "Data words", page 6 , representing the number of pixels in one line.

Data [4]...Data [7] = wlines = unsigned word (2bytes), representing the number of lines.

for xga resolution panels : wpixels = 1024 ; wlines = 768 lines

3.34 Logo status, Read

Description

Reads the status of the logo.

Command

Command [0]	\xf1
Command [1]	\x01

Returned Data

Data [0] =byStatus = 1 byte representing the status of the logo

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	-	saving	valid	fullscreen active

Table 3-289
byStatus byte description

3.35 Take screenshot (Logo)

Description

Takes a screenshot of the projected image for the logo, the logo image information is stored in an SDRAM.

Command

Command [0]	\xf1
Command [1]	\x86

Example

Takes a screenshot

Transmit	
Start	\xfe
Projector address	\x01
Command[0]	\xf1
Command[1]	\x86
Checksum	\x78
Stop	\xff

Receive (acknowledge)	
Start	\xfe
Projector address	\x01
Command[0]	\x00
Command[1]	\x06

Receive (acknowledge)	
Checksum	\x07
Stop	\xff

3.36 Save screenshot (Logo)

Description

Saves the previously taken screenshot (see the "take screenshot"command, "Take screenshot (Logo)", page 160)from the SDRAM to a flash memory.
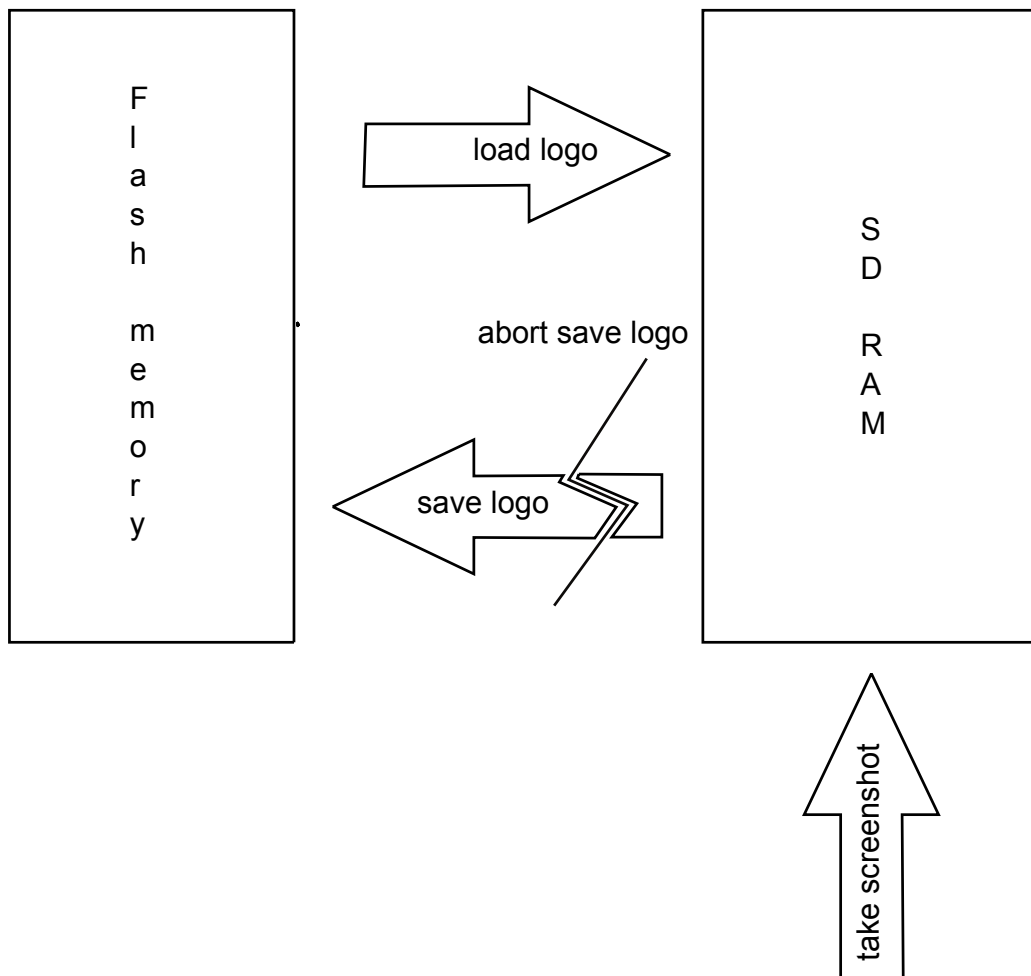
Image 3-11

Command

Command [0]	\xf1
Command [1]	\x87

3.37 Abort saving screenshot (Logo)

Description

Aborts saving the previously taken screenshot.
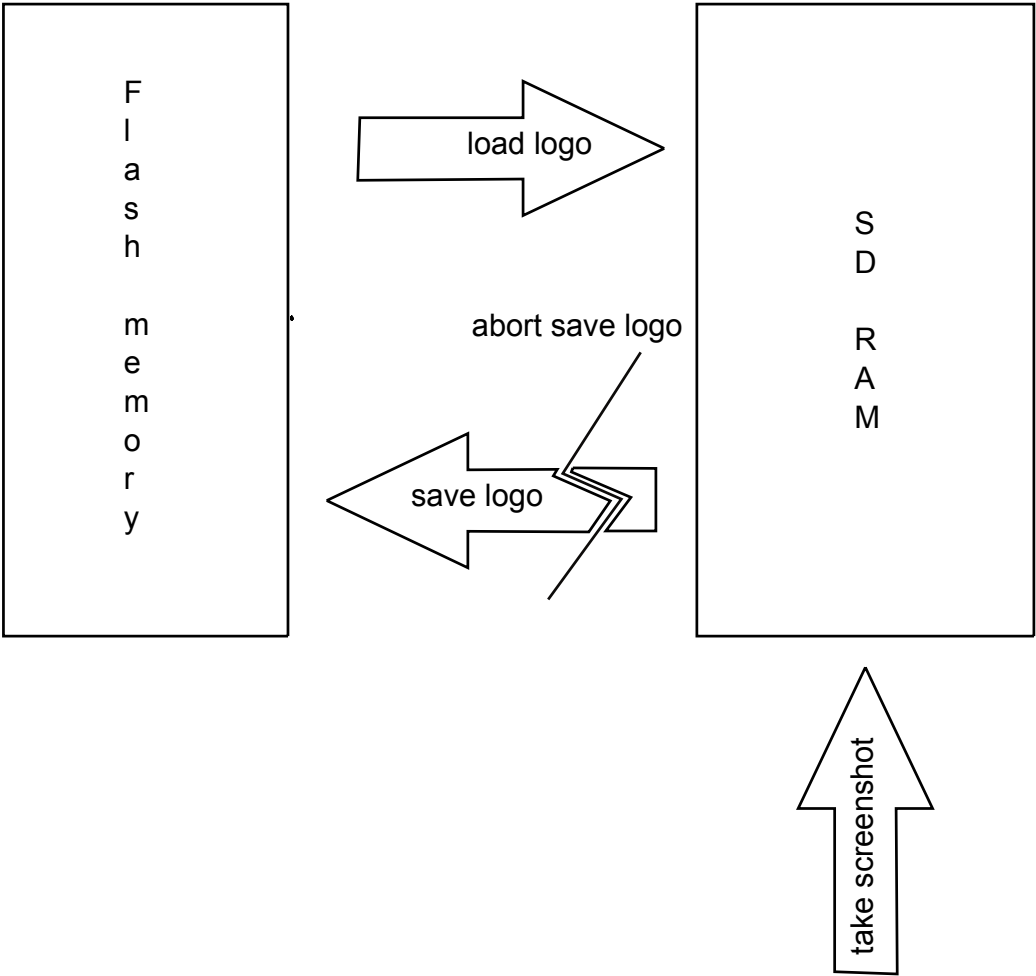
Image 3-12

Command

Command [0]	\xf1
Command [1]	\x88

3.38 Load logo

Description

Loads the logo (present in the flash memory) in the SDRAM.
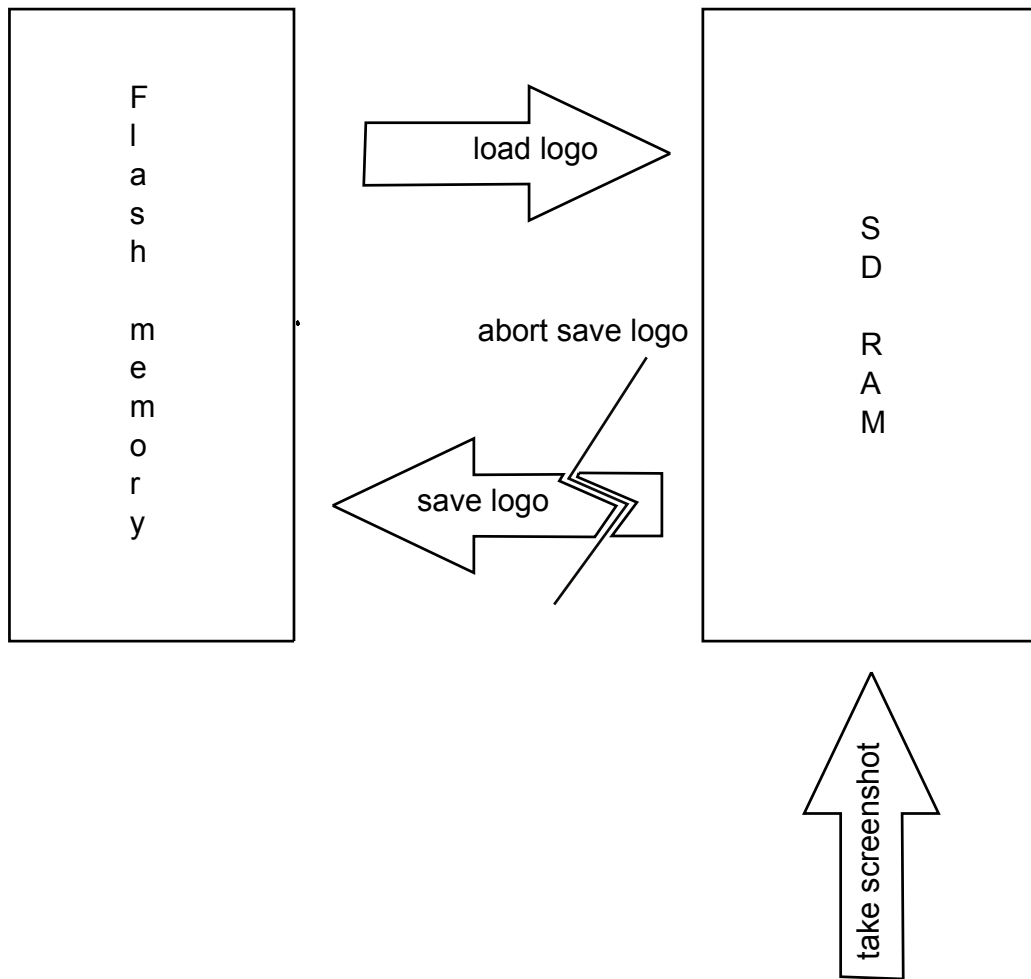
Image 3-13

Command

Command [0]	\xf1
Command [1]	\x88

4. RCU COMMANDS

4.1 Overview

Introduction

The RCU keys can be simulated by using the RCU command `\x30` (see *RCU commands* command)

followed by one of the following codes.

> **Key simulation will only work if the function is supported by the firmware.**

RCU

Button	Command
0	`\x19`
1	`\x10`
2	`\x11`
3	`\x12`
4	`\x13`
5	`\x14`
6	`\x15`
7	`\x16`
8	`\x17`
9	`\x18`
Standby	`\x0e`
Mute	`\x1f`
Text	`\x0d`
Volume "+"	`\x38`
Volume "-"	`\x39`
Brightness	`\x27`
Brightness "+"	`\x2a`
Brightness "-"	`\x2b`
Contrast	`\x25`
Contrast "-"	`\x28`
Contrast "+"	`\x29`
Color	`\x30`
Color "+"	`\x2c`
Color "-"	`\x2d`
Bass "+"	`\x3a`
Bass "-"	`\x3b`

Button	Command
Treble "+"	\x3c
Treble "-"	\x3d
Balance +	\x3e
Balance -	\x3f
Tint	\x31
Tint "- "	\x2e
Tint "+"	\x2f
Pause	\x0f
Phase	\x32
Phase "+"	\x34
Phase "-"	\x35
Sharpness "+"	\x36
Sharpness "-"	\x37
↑	\x04
↓	\x05
Menu	\x09
Exit	\x08
Address	\x20
←	\x07
→	\x06
Enter	\x0a
Freeze	\x1b
F1	\x6b
F2	\x6c
F3	\x6d
F4	\x6e
F5	\x6f
Help	\x1e
Pause	\x0f
Back	\x08
Auto image	\x69
PiP select	\x58
Lens focus up	\x41
Lens focus down	\x42
Lens zoom up	\x43
Lens zoom down	\x44
Lens shift up	\x70

Button	Command
Lens shift down	\x71
Digital zoom up	\x7d
Digital zoom down	\x7e
Logo	\x78
PiP adj	\x68
Lens adj	\x77

Revision Sheet

To:

▶ **Barco nv Events/Documentation**
Noordlaan 5, B-8520 Kuurne
Phone: +32 56.36.89.70, Fax: +32 56.36.88.24
E-mail: service.media_and_entertainment.emea@barco.com, Web: www.barco.com

From: _____

Date: _____

Please correct the following points in this documentation (**R5976884/02**):

page	wrong	correct